Homework Task: Make a timesheet calculator

Task description: Make a timesheet calculator (Table) for each department using Python. The application takes as input the number of working days for the period and the employee's rate (part time). The timesheet is generated 2 times a month, on the 15th and 25th of the month. The output of the application should show the number of hours per working day that each employee in the department has worked.

1. Process of creating code

Choose the right module:

```
import xlrd
import xlwt
import tkinter as tk
```

Define Function 'create_table()' to facilitate automatic creation of schedules as input:

```
def create_table():
    workbook = xlwt.Workbook(encoding= 'ascii')
    worksheet = workbook.add_sheet("Sheet1")
    worksheet.write(0,0, "Name")
    worksheet.write(0,1, "Department/Proffesion")
    for i in range(31):
        worksheet.write(0,2+i,str(i+1)+' day')
    worksheet.write(0,33, "Days/hours from 1 to 15")
    worksheet.write(0,34, "Day/hours from 1 to 31")
    workbook.save("timesheet_input.xls")
```

The file will be saved with the name "timesheet_input.xls"

Create a dictionary for querying part-time and department:

```
Create a mapping between position and time

Create a mapping between positions and departments

part_time = {'W_Tr':0.25,'W_Tr ':0.25,'H_Tr':0.25,'K_Tr':0.2,'M_Tr':0.1,'C_O':0.25,'W_1_O':0.25,'W_2_O':0.25}
department = {'W_Tr':'Tr','W_Tr ':'Tr','H_Tr':'Tr','K_Tr':'Tr','M_Tr':'Tr','C_O':'o','W_1_O':'o','W_2_O':'o'}
```

Define function 'get_result()', read input, and calculate output:

```
def get_result(row):
    name = datasheet.cell_value(row,0)
    pos = datasheet.cell_value(row,1)
    rowdata=datasheet.row_values(row,2,33)
    rate = part_time[pos]
    dep = department[pos]
    day15 = 0
    day31 = 0
    for i in range(15):
        if rowdata[i]=='0':
            day15+=1
    for j in range(31):
        if rowdata[j]=='0':
            day31+=1
    result = ('Employee\'s name: ' + str(name) + '\n'
              'Department: ' + str(dep) + '\n'
              'In 15 days: ' + str(day15) + ' Days' + '/' + str(rate*day15*8) + ' hours' + '\n'
              'In a month: ' + str(day31) + ' Days' + '/' + str(rate*day31*8) + ' hours')
    return result
```

Define the function 'show_result' to display the result in the window:

```
def show_result(name):
    namelist = datasheet.col_values(0)
    try:
        index = namelist.index(name)
        text = get_result(index)
        return text
    except:
        error = 'No employee found'
        return error
```

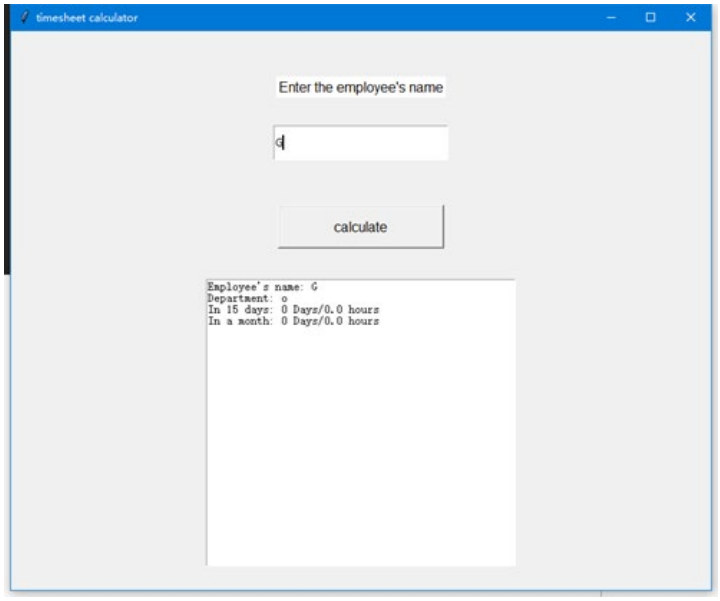
the result is in the form of a string

Create GUI window with Tkinter module:

```
window=tk.Tk()
window.title('timesheet calculator')
window.geometry('800x640')
text = tk.StringVar()
entry = tk.Entry(window,textvariable=text)
title = tk.Label(window,text='Enter the employee\'s name',bg='white',font=('Arial', 12))
title.place(relx=0.5,rely=0.1,anchor="center")
text.set('')
entry.place(relx=0.5,rely=0.2,anchor="center",height=40,width=200)
def printEntry():
    var= show_result(text.get())
    t.delete("1.0","end")
    t.insert('end', var)
button = tk.Button(window,text='calculate',command=printEntry,height=2,width=20,font=('Arial', 12))
button.place(relx=0.5,rely=0.35,anchor="center")
t = tk.Text(width=50, height=25)
t.place(relx=0.5,rely=0.7,anchor="center")
window.mainloop()
```

All components are centered

Debugging the program, the result runs successfully:



2. Create an executable file under the windows system for the program

Import os and sys modules

In order to avoid errors in reading the path of the xls file after the exe is generated, use the function "os.path.dirname(os.path.realpath(sys. executable)" to read the root directory where the py.exe file is located

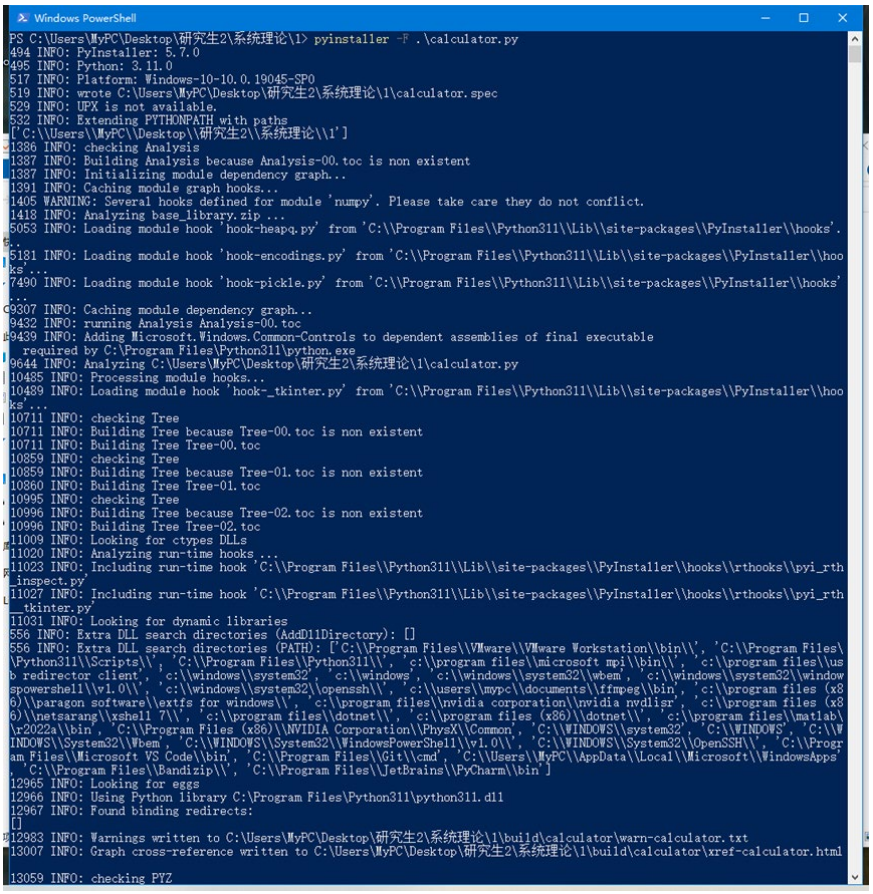
```
import xlrd
import xlwt
import tkinter as tk
import os
import sys

filepath = os.path.join(os.path.dirname(os.path.realpath(sys.executable)),'timesheet_input.xls')
```

Define the main function

Use sys.argv[] Reads the incoming parameters at runtime:

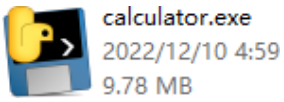
```
def main():
    window.mainloop()

if __name__ == '__main__':
    try:
        if sys.argv[1] == 'create':
            create_table()
        elif sys.argv[1] == 'calculate':
            main()
        else:
            print('The parameter is wrong, please enter \'create\'to create the timesheet')
    except:
        main()
```

Use the pyinstaller tool to generate an executable file exe, Enter the command in the shell: "pyinstaller -F .\calculator.py"

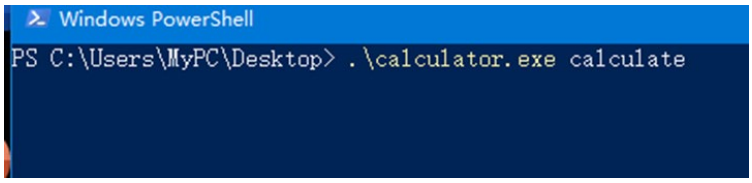


3. Check run results

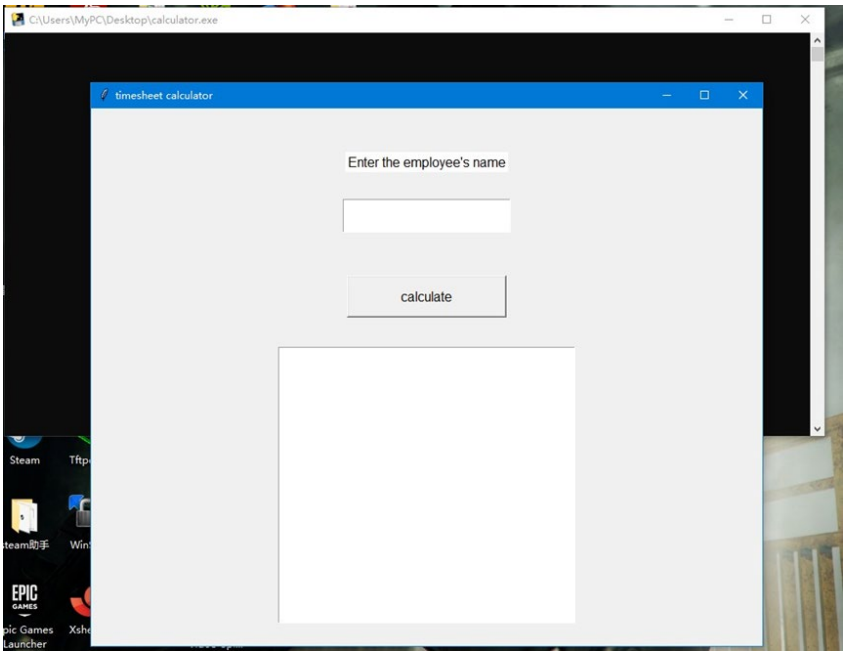
Now get the file:



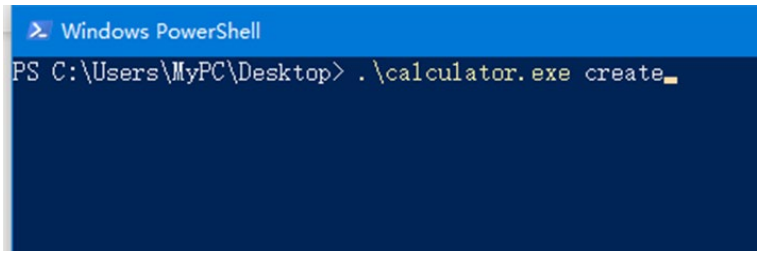
Substitute the parameter “calcuate” to run:



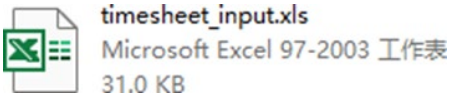
The calculator GUI interface start directly, which is equivalent to double-clicking to run:



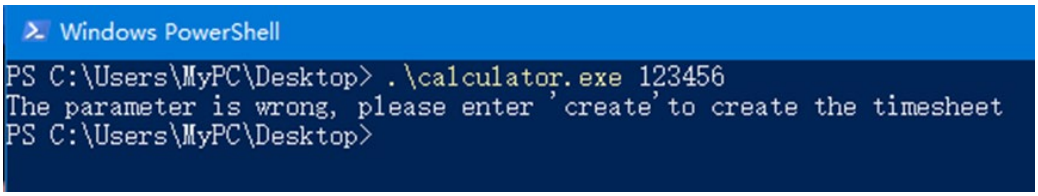
Substitute the parameter “create” to run:



A blank timetable xls file is created automatically:



If enter the wrong parameters at runtime:



Prompt appears: parameter error

4. Manual for Users

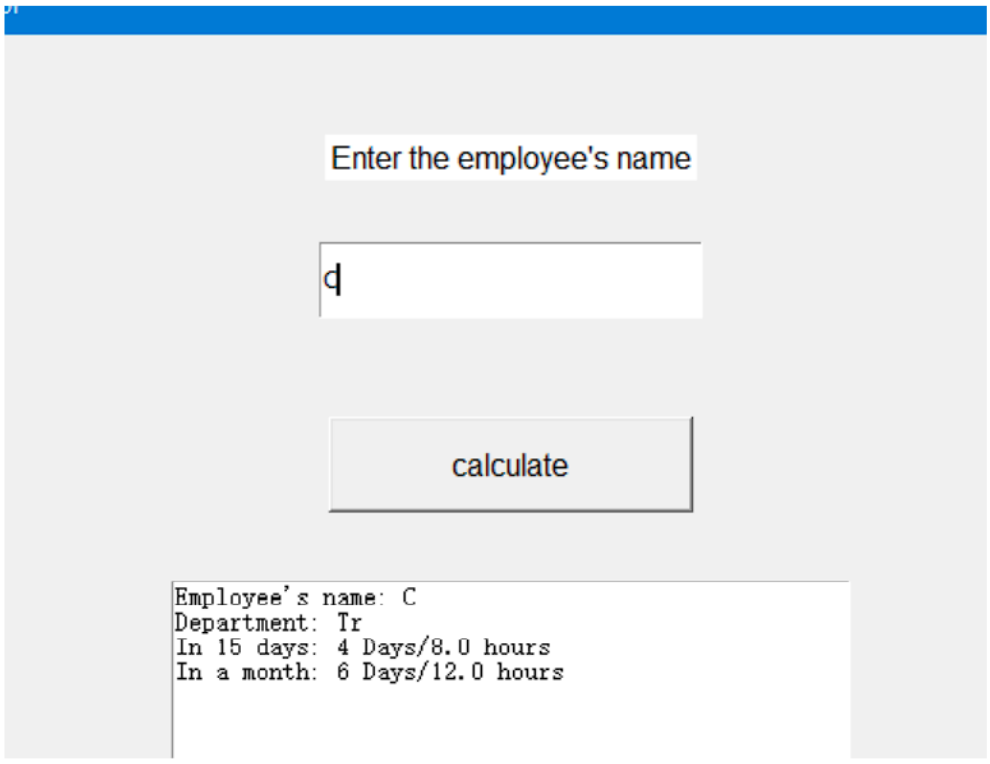
First, use the command ".\calculator.exe create" to create a new file named “timesheet_input.xls”

```
PS C:\Users\MyPC> .\calculator.exe create
```

Then manually register the employee's work information in the time sheet:

	A	B	C	D	E	F	G	H	I	J	K	L	
1	Name	Department/Proffesion	1 day	2 day	3 day	4 day	5 day	6 day	7 day	8 day	9 day	10 day	11
2	A	Ш_Тр	O	O									
3	B	Ш_Тр			O	O	O						
4	C	Н_Тр						O	O	O	O		
5	D	К_Тр								O		O	
6	E	М_Тр											
7	F	С_О				O		O					
8	G	Ш_1_О											
9	H	Ш_2_О		O		O				O			

Make sure that the “ timesheet_input.xls” file and the exe executable file are in the same directory, start the calculator, enter the name of the employee, and the calculation result will be obtained and displayed.



Conclusion

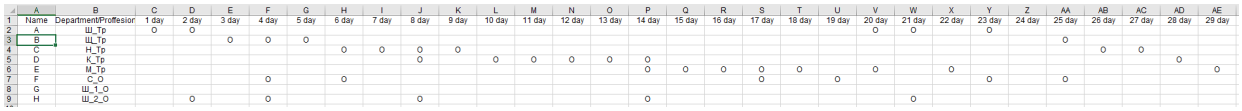
I learned the basic programming operations of python, understood the data processing and operation. And use Tkinter module to understand the design and interaction principle of GUI interface. A user manual is created.

According to the test results:

- The program runs normally without error.
- xls files are created and read normally
- Calculations are displayed correctly

Appendix A

timesheet_input.xls



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE
1	Name	Department/Proffesion	1 day	2 day	3 day	4 day	5 day	6 day	7 day	8 day	9 day	10 day	11 day	12 day	13 day	14 day	15 day	16 day	17 day	18 day	19 day	20 day	21 day	22 day	23 day	24 day	25 day	26 day	27 day	28 day	29 day
2	A	Ш_Тр	O	O																		O	O		O						
3	B	ЦЦ_Тр			O	O	O																				O				
4	C	Н_Тр						O	O	O	O																	O	O		
5	D	К_Тр								O		O	O	O	O	O														O	
6	E	М_Тр														O	O	O	O	O		O		O							O
7	F	С_О				O		O											O		O				O		O				
8	G	Ш_1_О																													
9	H	Ш_2_О		O		O				O						O							O								

Appendix B

source code

```
1.  import xlrd
2.  import xlwt
3.  import tkinter as tk
4.  import os
5.  import sys
6.
7.  # filepath = os.path.join(os.path.dirname(os.path.realpath(sys.executable)),
    'timesheet_input.xls')
8.  filepath = './timesheet_input.xls'
9.
10. def create_table():
11.     workbook = xlwt.Workbook(encoding= 'ascii')
12.     worksheet = workbook.add_sheet("Sheet1")
13.     worksheet.write(0,0, "Name")
14.     worksheet.write(0,1, "Department/Proffesion")
15.     for i in range(31):
16.         worksheet.write(0,2+i,str(i+1)+' day')
17.     worksheet.write(0,33, "Days/hours from 1 to 15")
18.     worksheet.write(0,34, "Day/hours from 1 to 31")
19.     workbook.save("timesheet_input.xls")
20.
```

```
part_time = {'Ш_Tp':0.25,'Ш_Tp ':0.25,'H_Tp':0.25,'K_Tp':0.2,'M_Tp':0.1,'C_O':0.25,'Ш_1_O':0.25,'Ш_2_O':0.25}
department = {'Ш_Tp':'Tr','Ш_Tp ':'Tr','H_Tp':'Tr','K_Tp':'Tr','M_Tp':'Tr','C_O':'o','Ш_1_O':'o','Ш_2_O':'o'}

data = xlrd.open_workbook(filepath)
datasheet = data.sheets()[0]
nrow = datasheet.nrows
ncol = datasheet.ncols

def get_result(row):
    name = datasheet.cell_value(row,0)
    pos = datasheet.cell_value(row,1)
    rowdata=datasheet.row_values(row,2,33)
    rate = part_time[pos]
    dep = department[pos]
    day15 = 0
    day31 = 0
    for i in range(15):
        if rowdata[i]=='O':
            day15+=1
    for j in range(31):
        if rowdata[j]=='O':
            day31+=1
    result = ('Employee\'s name: ' + str(name) + '\n'
              'Department: ' + str(dep) + '\n'
              'In 15 days: ' + str(day15) + ' Days' + '/' + str(rate*day15*8) + ' hours' + '\n'
              'In a month: ' + str(day31) + ' Days' + '/' + str(rate*day31*8) + ' hours')
    return result

def show_result(name):
    namelist = datasheet.col_values(0)
    try:
        index = namelist.index(name)
        text = get_result(index)
        return text
    except:
        error = 'No employee found'
        return error

window=tk.Tk()
window.title('timesheet calculator')
```

```
window.geometry('800x640')
text = tk.StringVar()
entry = tk.Entry(window,textvariable=text)
title = tk.Label(window,text='Enter the employee\'s name',bg='white',font=('Arial', 12))
title.place(relx=0.5,rely=0.1,anchor="center")
text.set('')
entry.place(relx=0.5,rely=0.2,anchor="center",height=40,width=200)
t = tk.Text(width=50, height=25)
t.place(relx=0.5,rely=0.7,anchor="center")
def printEntry():
    var = show_result(text.get())
    t.delete("1.0","end")
    t.insert('end', var)
button = tk.Button(window,text='calculate',command=printEntry,height=2,width=20,font=('Arial', 12))
button.place(relx=0.5,rely=0.35,anchor="center")

def main():
    window.mainloop()

if __name__ == '__main__':
    try:
        if sys.argv[1] == 'create':
            create_table()
        elif sys.argv[1] == 'calculate':
            main()
        else:
            print('The parameter is wrong, please enter \'create\'to create the timesheet')
    except:
        main()
```