

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_Tp':0.25,'W_Tp ':0.25,'H_Tp':0.25,'K_Tp':0.2,'H_Tp':0.1,'G_B':0.25,'W_1_B':0.25,'W_2_B':0.25}
department = {'W_Tp': 'Tr','W_Tp ': 'Tr','H_Tp': 'Tr','K_Tp': 'Tr','H_Tp': 'Tr','G_B': 'o','W_1_B': 'o','W_2_B': '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:

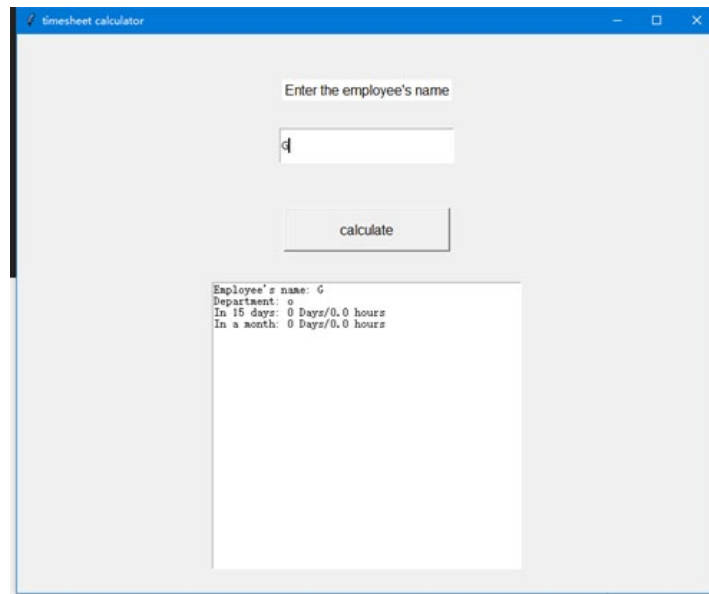
```

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,relx=0.1,anchor="center")
text.set('')
entry.place(relx=0.5,relx=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,relx=0.35,anchor="center")
t = tk.Text(width=50, height=25)
t.place(relx=0.5,relx=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()

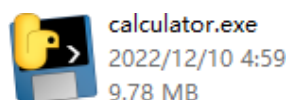
if __name__ == '__main__':
    try:
        if sys.argv[1] == 'create':
            create_table()
        elif sys.argv[1] == 'calculate':
            main()
        else:
            print('The parameter is wrong, please enter \'create\' to create the timesheet')
    except:
        main()
```

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

```
Windows PowerShell
PS C:\Users\MyPC\Desktop\研究生2\系统理论\1> pyinstaller -F .\calculator.py
494 INFO: PyInstaller: 5.7.0
495 INFO: Python: 3.11.0
517 INFO: Platform: Windows-10-10.0.19045-SFO
519 INFO: wrote C:\Users\MyPC\Desktop\研究生2\系统理论\1\calculator.spec
529 INFO: UFX is not available.
532 INFO: Extending PYTHONPATH with paths
['C:\Users\MyPC\Desktop\研究生2\系统理论\1']
1386 INFO: checking Analysis
1387 INFO: Building Analysis because Analysis-00.toc is non existent
1387 INFO: Initializing module dependency graph...
1391 INFO: Caching module graph hooks...
1405 WARNING: Several hooks defined for module 'numpy'. Please take care they do not conflict.
1418 INFO: Analyzing base_library.zip ...
5053 INFO: Loading module hook 'hook-heapq.py' from 'C:\Program Files\Python311\Lib\site-packages\PyInstaller\hooks'.
5181 INFO: Loading module hook 'hook-encodings.py' from 'C:\Program Files\Python311\Lib\site-packages\PyInstaller\hooks'.
7490 INFO: Loading module hook 'hook-pickle.py' from 'C:\Program Files\Python311\Lib\site-packages\PyInstaller\hooks'.
9307 INFO: Caching module dependency graph...
9432 INFO: running Analysis Analysis-00.toc
9439 INFO: Adding Microsoft.Windows.Common-Controls to dependent assemblies of final executable
required by C:\Program Files\Python311\python.exe
9644 INFO: Analyzing C:\Users\MyPC\Desktop\研究生2\系统理论\1\calculator.py
10489 INFO: Processing module hooks...
10489 INFO: Loading module hook 'hook-_tkinter.py' from 'C:\Program Files\Python311\Lib\site-packages\PyInstaller\hooks'.
10711 INFO: checking Tree
10711 INFO: Building Tree because Tree-00.toc is non existent
10711 INFO: Building Tree Tree-00.toc
10859 INFO: checking Tree
10859 INFO: Building Tree because Tree-01.toc is non existent
10860 INFO: Building Tree Tree-01.toc
10995 INFO: checking Tree
10996 INFO: Building Tree because Tree-02.toc is non existent
10996 INFO: Building Tree Tree-02.toc
11009 INFO: Looking for ctypes DLLs
11020 INFO: Analyzing run-time hooks ...
11023 INFO: Including run-time hook 'C:\Program Files\Python311\Lib\site-packages\PyInstaller\hooks\rthooks\pyi_rth_inspect.py'
11027 INFO: Including run-time hook 'C:\Program Files\Python311\Lib\site-packages\PyInstaller\hooks\rthooks\pyi_rth_tkinter.py'
11031 INFO: Looking for dynamic libraries
556 INFO: Extra DLL search directories (AddDllDirectory): []
556 INFO: Extra DLL search directories (PATH): ['C:\Program Files\VMware\VMware Workstation\bin\;', 'C:\Program Files\Python311\Scripts\;', 'C:\Program Files\Python311\;', 'c:\program files\microsoft mpi\bin\;', 'c:\program files\usb redirector client\;', 'c:\windows\system32\;', 'c:\windows\c:\windows\system32\wbem\;', 'c:\windows\system32\window powershell\v1.0\;', 'c:\windows\system32\openssh\;', 'c:\users\lmyc\documents\ffmpeg\bin\;', 'c:\program files (x86)\paragon software\extfs for windows\;', 'c:\program files\nvidia corporation\nvidia nvdlisr\;', 'c:\program files (x86)\netsarang\xshell 7\;', 'c:\program files\dotnet\;', 'c:\program files (x86)\dotnet\;', 'c:\program files\matlab\az2022a\bin\;', 'C:\Program Files (x86)\NVIDIA Corporation\PhysX\Common\;', 'C:\WINDOWS\system32\;', 'C:\WINDOWS\;', 'C:\WINDOWS\System32\Wbem\;', 'C:\WINDOWS\System32\WindowsPowerShell\v1.0\;', 'C:\WINDOWS\System32\OpenSSH\;', 'C:\Program Files\Microsoft VS Code\bin\;', 'C:\Program Files\Git\cmd\;', 'C:\Users\MyPC\AppData\Local\Microsoft\WindowsApps\;', 'C:\Program Files\Bandizip\;', 'C:\Program Files\JetBrains\FyCharm\bin\']
12965 INFO: Looking for eggs
12966 INFO: Using Python library C:\Program Files\Python311\python311.dll
12967 INFO: Found binding redirects:
[]
12983 INFO: Warnings written to C:\Users\MyPC\Desktop\研究生2\系统理论\1\build\calculator\warn-calculator.txt
13007 INFO: Graph cross-reference written to C:\Users\MyPC\Desktop\研究生2\系统理论\1\build\calculator\lxfref-calculator.html
13059 INFO: checking PTZ
```

3. Check run results

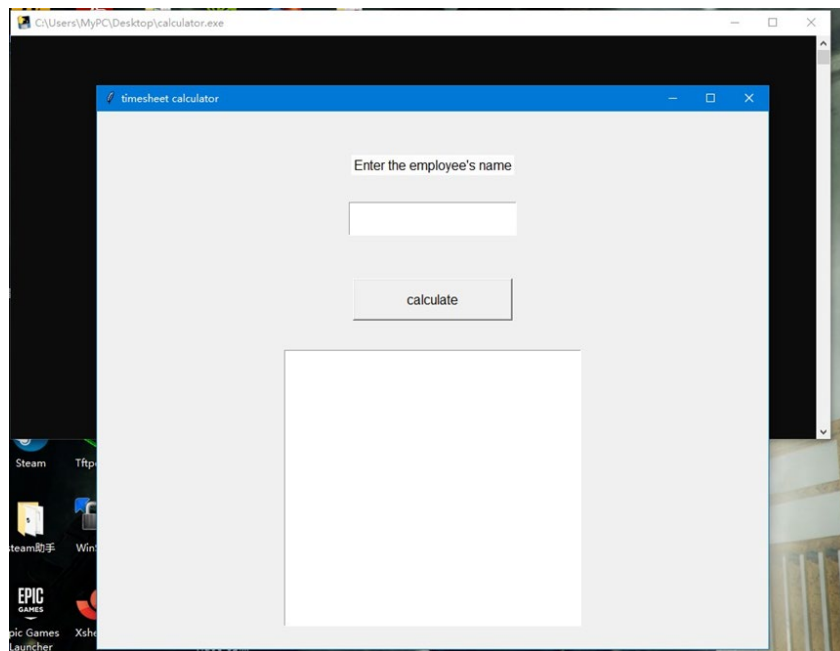
Now get the file:



Substitute the parameter “calculate” to run:

```
Windows PowerShell
PS C:\Users\MyPC\Desktop> .\calculator.exe calculate
```

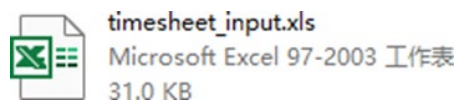
The calculator GUI interface start directly, which is equivalent to double-clicking to run:



Substitute the parameter "create" to run:

```
Windows PowerShell
PS C:\Users\MyPC\Desktop> .\calculator.exe create_
```

A blank timetable xls file is created automatically:



If enter the wrong parameters at runtime:

```
Windows PowerShell
PS C:\Users\MyPC\Desktop> .\calculator.exe 123456
The parameter is wrong, please enter 'create' to create the timesheet
PS C:\Users\MyPC\Desktop>
```

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/Profession	1 day	2 day	3 day	4 day	5 day	6 day	7 day	8 day	9 day	10 day	11
2	A	W_Tp	0	0									
3	B	W_Tp			0	0	0						
4	C	H_Tp						0	0	0	0		
5	D	K_Tp								0		0	
6	E	M_Tp											
7	F	C_O				0		0					
8	G	W_1_O											
9	H	W_2_O		0		0				0			

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.

Enter the employee's name

d

calculate

Employee's name: C
Department: Tr
In 15 days: 4 Days/8.0 hours
In a month: 6 Days/12.0 hours

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/Profession	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	0	0																		0	0								
3	B	Ш.Тр			0	0	0																				0	0	0		
4	C	Ш.Тр						0	0	0	0		0	0	0	0													0	0	
5	D	Ш.Тр											0	0	0	0	0	0	0	0	0			0	0					0	
6	E	Ш.Тр																							0	0					
7	F	Ш.Тр																									0				
8	G	Ш.Тр																													
9	H	Ш.Тр		0		0				0						0						0									

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)),
8. # 'timesheet_input.xls')
9. filepath = './timesheet_input.xls'
10.
11. def create_table():
12.     workbook = xlwt.Workbook(encoding= 'ascii')
13.     worksheet = workbook.add_sheet("Sheet1")
14.     worksheet.write(0,0, "Name")
15.     worksheet.write(0,1, "Department/Profession")
16.     for i in range(31):
17.         worksheet.write(0,2+i, str(i+1)+ ' day')
18.     worksheet.write(0,33, "Days/hours from 1 to 15")
19.     worksheet.write(0,34, "Day/hours from 1 to 31")
20.     workbook.save("timesheet_input.xls")
```

```

21. part_time = {'W_Tp':0.25,'W_Tp ':0.25,'H_Tp':0.25,'K_Tp':0.2,'M_Tp':0.1,'C_0
    ':0.25,'W_1_0':0.25,'W_2_0':0.25}
22. department = {'W_Tp':'Tr','W_Tp ':'Tr','H_Tp':'Tr','K_Tp':'Tr','M_Tp':'Tr','
    C_0':'o','W_1_0':'o','W_2_0':'o'}
23.
24. data = xlrd.open_workbook(filepath)
25. datasheet = data.sheets()[0]
26. nrow = datasheet.nrows
27. ncol = datasheet.ncols
28.
29. def get_result(row):
30.     name = datasheet.cell_value(row,0)
31.     pos = datasheet.cell_value(row,1)
32.     rowdata=datasheet.row_values(row,2,33)
33.     rate = part_time[pos]
34.     dep = department[pos]
35.     day15 = 0
36.     day31 = 0
37.     for i in range(15):
38.         if rowdata[i]=='0':
39.             day15+=1
40.     for j in range(31):
41.         if rowdata[j]=='0':
42.             day31+=1
43.     result = ('Employee\'s name: ' + str(name) + '\n'
44.             'Department: ' + str(dep) + '\n'
45.             'In 15 days: ' + str(day15) + ' Days' + '/' + str(rate*day15*8
46.             ) + ' hours' + '\n'
47.             'In a month: ' + str(day31) + ' Days' + '/' + str(rate*day31*8
48.             ) + ' hours')
49.     return result
50.
51. def show_result(name):
52.     namelist = datasheet.col_values(0)
53.     try:
54.         index = namelist.index(name)
55.         text = get_result(index)
56.         return text
57.     except:
58.         error = 'No employee found'
59.         return error
60.
61. window=tk.Tk()
62. window.title('timesheet calculator')

```



```

61. window.geometry('800x640')
62. text = tk.StringVar()
63. entry = tk.Entry(window,textvariable=text)
64. title = tk.Label(window,text='Enter the employee\'s name',bg='white',font=('
    Arial', 12))
65. title.place(relx=0.5,rely=0.1,anchor="center")
66. text.set('')
67. entry.place(relx=0.5,rely=0.2,anchor="center",height=40,width=200)
68. t = tk.Text(width=50, height=25)
69. t.place(relx=0.5,rely=0.7,anchor="center")
70. def printEntry():
71.     var = show_result(text.get())
72.     t.delete("1.0","end")
73.     t.insert('end', var)
74. button = tk.Button(window,text='calculate',command=printEntry,height=2,width
    =20,font=('Arial', 12))
75. button.place(relx=0.5,rely=0.35,anchor="center")
76.
77. def main():
78.     window.mainloop()
79.
80. if __name__ == '__main__':
81.     try:
82.         if sys.argv[1] == 'create':
83.             create_table()
84.         elif sys.argv[1] == 'calculate':
85.             main()
86.         else:
87.             print('The parameter is wrong, please enter \'create\' to create
                the timesheet')
88.     except:
89.         main()

```