



河南理工大学

《Python 程序设计》

课程设计报告

(2018 —2019 学年第 一 学期)

题 目 _____ 简易计算器 _____

学生姓名 _____ 郭乐乐 _____

专业班级 _____ 信管 1601 _____

学生学号 _____ 311609030115 _____

教师姓名 _____ 徐 文 鹏 _____

成 绩:

评 语:

教师签名:

日期:

目录

一、设计目的.....	3
二、设计任务和要求.....	3
三、总体设计.....	4
三、图形化界面详细设计.....	6
四、设计结果与分析.....	8
五、详细设计.....	9
六、调试与测试.....	11
七、设计总结.....	12
八、参考文献.....	13
九、源程序.....	13

一、设计目的

1. 课程设计目的

为了进一步提高我们的逻辑思维能力，合作动手能力以及独立解决问题能力，我们选择了“小计算器”课题来进行 PYTHON 语言的强化训练。

1. 通过使用小计算器可以使复杂困难的计算变得简单，从而提高计算的准确率。

2. 作为计算机专业的学生，经常通过实验研究各种课题，有许多的数据需要记录、计算和处理，小计算器使实验数据的处理变得更加便捷、精准。

3. 计算器为数学应用提供了先进的计算工具，更便于处理实际数据，特别是处理随机实验得来的数据，使数学应用有了广阔的空间。

4. 计算是认识客观世界最基本的工具，是培养学生思维能力的载体，是科学技术的载体。

5. 小计算器可以读数和计数、知道时间、购物付款和找零、计重和测量，以此帮助人们看懂浅显易懂的时间表及简单的图表和图示，以及完成与此相关的必要计算、估算和近似计算。

二、设计任务和要求

1. 设计任务

1. 设计的计算器应用程序可以完成加法、减法、乘法、除法以及取余运算（可以进行浮点数和负数的运算）

2. 有求倒数、退格和清零功能。

3. 有三角函数，开根，平方，对数运算

2. 设计要求

1. 设计的计算器应用程序可以完成加法、减法、乘法、除法以及取余（可以进行浮点数和负数的运算）三角函数，开根，平方，对数运算；要求有退格和清零功能。

2. 课程设计可选用 PyCharm, Python IDLE 等开发平台以提高开发效率，通过资料查阅和学习尽可能掌握一种或多种集成开发环境。

3. 团队合作完成计算器功能，并分组负责对应文档编写，课程设计报告内容包括：设计目的、设计要求、总体设计、详细设计、调试与测试、设计总结等六方面。

3. 设计效果图

设计时先在设计界面中添加按钮等控件, 后通过函数主流程加入了按钮功能, 主要通过 Python3+和 Python GUI tkinter 语法, 利用了 Python 内置的 math 库来进行三角函数等的运算

三、总体设计

1. 主要完成简单的计算器，实现加减乘除，三角函数，开根，平方，对数运算，清除，回退，显示计算过程

2. 总体上基于 python3+, python GUI Tkinter 语法编程

3. 分为

- ① 图形化界面，界面美化，pyinstaller 打包
- ② 按钮功能设计，函数主流程，实现加减乘除，回退，清除功能
- ③ QA 测试并提出对现有功能的完善方案，增加三角函数，开根，平方，对数运算

1. 负责分工

程龙：图形化界面，界面美化，pyinstaller 打包

郭乐乐：按钮功能设计，函数主流程，实现加减乘除，回退，清除功能

马志豪：测试并提出对现有功能的完善，增加三角函数，平方，对数运算

2. 功能实现

1) 加减乘除括号运算

```
def buttonClick(btn):  
    content = result2.get()  
    a=''  
    if btn in '()e%123+456-789*.0/,':  
        content += btn  
    elif btn in '←':  
        content =content[:-1]  
    elif btn == 'π':  
        content += 'pi'  
    elif btn == 'sin':  
        content += 'sin('  
    elif btn == 'cos':  
        content += 'cos('  
    elif btn == 'AC':  
        content = ''  
    elif btn == '^2':  
        content += '**2'  
    elif btn == 'log':  
        content += 'log('  
    elif btn == '√':  
        content += 'sqrt('  
    elif btn == '=':  
        a = '='+str(eval(content))  
    result2.set(content)  
    result.set(a[:15])
```

函数主流程实现的按钮功能主要通过 eval 进行运算处理。

2) 其他运算

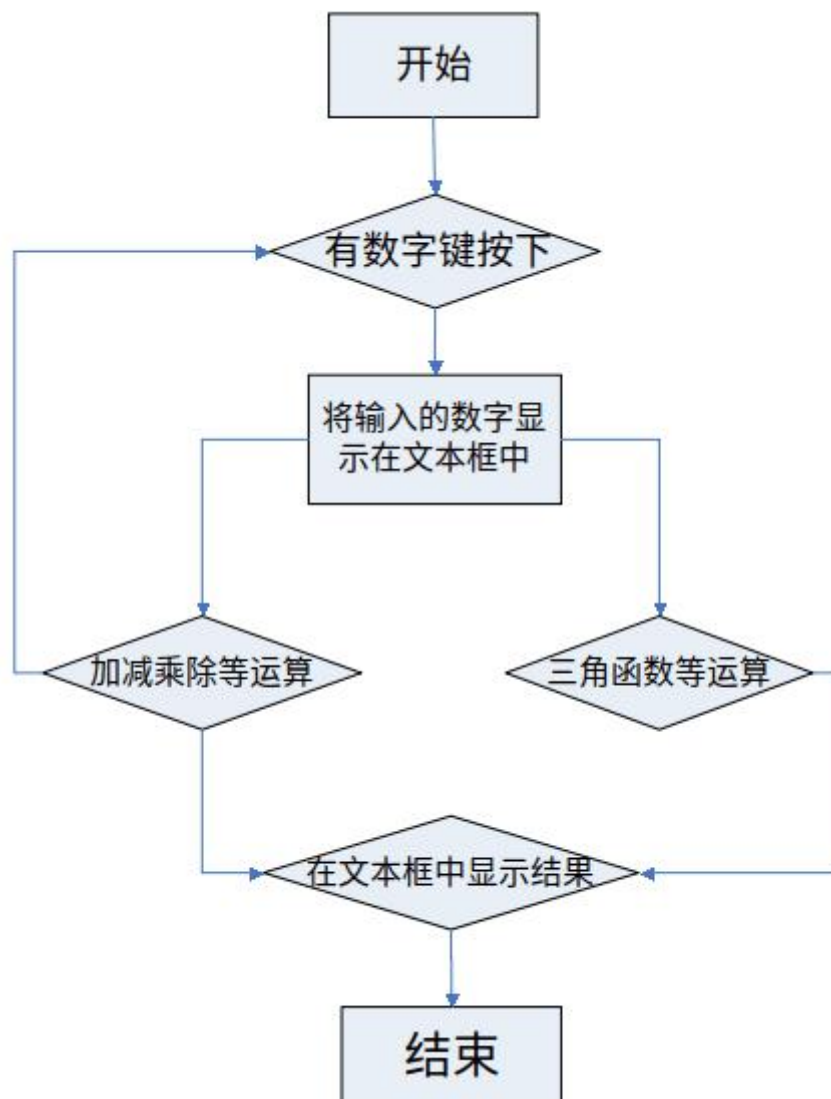
对于三角函数运算,开根,平方,对数运算,主要用到了 Python 内置的 math

函数库，通过 eval 自动处理函数运算结果实现。例如求 $\sin \pi$ 。则是

```
import math  
  
math.sin(math.pi)
```

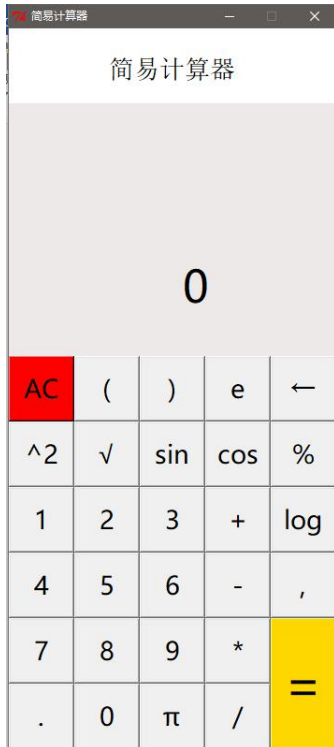
这里用到了 math 来帮助我们求得运算结果，我们统一交给 eval 处理。

3) 函数流程图



三、图形化界面详细设计

图形化界面主要用到了 Python GUI tkinter 语法进行设计，因为该库是 python 内置的 GUI 函数库，所以我们不需要安装就可以 import 直接使用



```

from tkinter import *

root = Tk()

root.title('简易计算器')

root.geometry('350x770')

root.resizable(width=False,height=False)

l = Label(root, text="简易计算器",
bg="white",fg='black',font=("Arabic", 20),
width=400, height=3)

l.pack(side=TOP)

label = Label(root,font = ('微软雅黑',20),bg = '#EEE9E9',bd = '9',fg =
'#828282',textvariable = result2)

label.place(x=0,y=80,width = 350,height = 200)

label2 = Label(root,font = ('微软雅黑',35),bg = '#EEE9E9',bd = '9',fg
= 'black',textvariable = result)

label2.place(x=0,y=200,width = 400,height =150)

#16 个符号
number=['AC','(','')','e','←','^2','√','sin','cos','%','1','2','
3','+', 'log','4','5','6',' ',' ',' ','7','8','9','*',' ','.', '0','π','/','
']

index = 0

for row in range(6):
    for col in range(5):
        num = number[index]

        index+=1

        btnDight = Button(root,text=num,font = ('微软雅黑',20),fg =
('black'),command=lambda x=num:buttonClick(x))

```

```

        btnDight.place(x=col*70,y=350+row*70,width=70,height=70)

# 等号美化
    btnDight = Button(root,text='=',font = ('微软雅黑',40),fg =
('black'),bg=('gold'),command=lambda x='':buttonClick(x))
    btnDight.place(x=280,y=330+300,width=70,height=140)

# AC 美化
    btn = Button(root,text='AC',font = ('微软雅黑',20),fg =
('Black'),bg=('red'),command=lambda x='AC':buttonClick(x))

    btn.place(x=0,y=350,width=70,height=70)

root.mainloop()

```

主要利用到了 tkinter 的 button 按钮控件在程序中显示按钮和 label 标签控件用于显示文本和位图，后来利用 menu 控件，message 控件等控件对图形化界面进行优化完善。

四、设计结果与分析

1.运行调试

1)进行加减乘除求余以及指数运算

带负数的浮点加法运算（例： $-1.2+4=2.8$ ）

依次按“+/-”（也可以最后按），“1.2”，“+”，“4”键

分析：计算器可以直接输入“-”，计算比较简便，带负号的浮点数加法做起来没有问题。

2)连续运算（例： $2*3*4=24.0$ ）

依次按下“3”、“*”、“4”、“/”、“6”、“=”键

3)清零

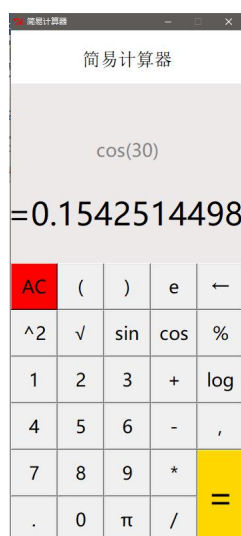
按下 AC 键清零

分析：按下清零键后数据消失

4)退格

输入数字按下退格键后可以回退一步操作

分析：按下退格清除一个数字



5) 求余弦、求正切、对 e 求幂、求平方根

例：cos30

先按下 cos 键，然后输入数字 30

尝试求对数，平方根，开根都可以得到正常的结果。

然而尝试 sin pi 却得到的不合常规的结果：

原因分析如下，因为使用 Python 内置的 math 进行计算，存在浮点数精度问题，在命令行尝试 `math.sin(math.pi)` 得到和计算器一样的结果，说明计算器通过该 math 库进行了这样的计算，导致存在了浮点数精度问题。简单的解决方案可以对输出结果进行处理。

五、详细设计

在整个课设小组进行设计的过程中，我主要负责可视化界面的按钮操作，在项目代码中，我通过设置两个小界面，来显示计算过程的输入与输出：

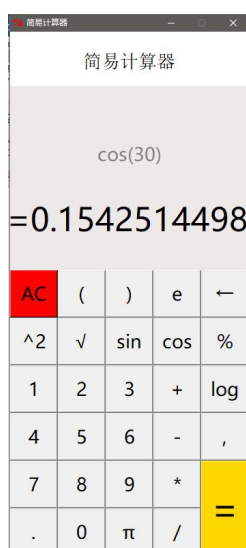
```
result = StringVar()
result.set('0')
result2 = StringVar()
result2.set('')
def buttonClick(btn):
    content = result2.get()
    a=''
    if btn in '()e%123+456-789*.0/,:':
        content += btn
    elif btn in '←':
        content =content[:-1]
    elif btn == 'π':
        content += 'pi'
    elif btn == 'sin':
        content += 'sin('
```

```

elif btn == 'cos':
content += 'cos('
elif btn == 'AC':
content = ''
elif btn == '^2':
content += '**2'
elif btn == 'log':
content += 'log('
elif btn == '√':
content += 'sqrt('
elif btn == '=':
a = '='+str(eval(content))
if content == 'sin(pi)':
content = 1
result2.set(content)
result.set(a[:15])

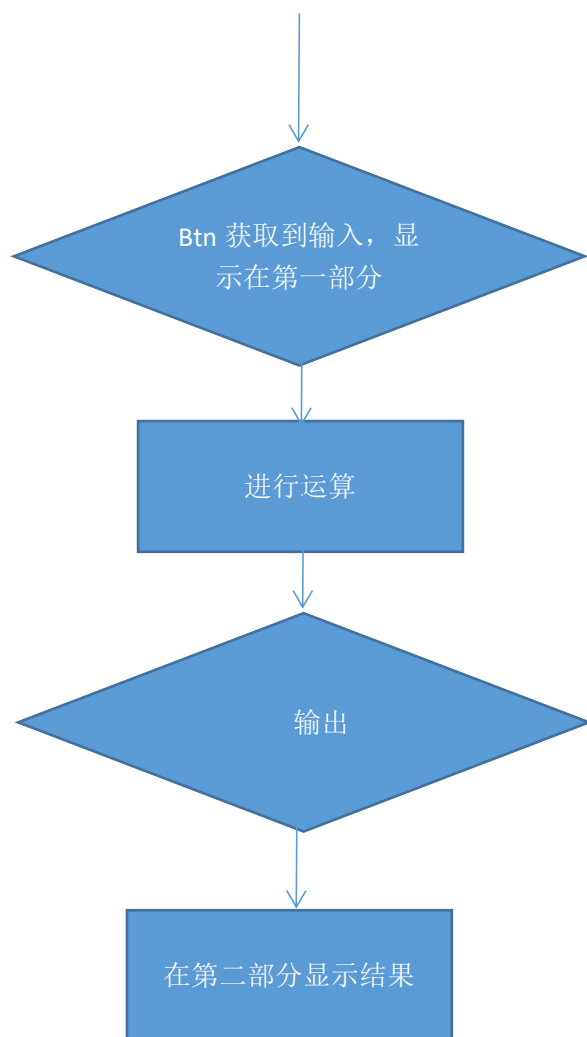
```

在默认显示界面上，我默认显示 0，当进行相关计算时（以下图为例），会在第一部分显示用户的输入：cos(30)，而在第二部分显示用户想要的得到的运算结果（0.15425）。



代码通过对用户按键操作的捕捉（btn 操作），可以将用户输入的数字或者运算符都存入 content 中，待得用户完成整个计算的输入，点击等于时，通过调用 eval 方法将 content 中的全部内容进行数值运算，而后装换成字符串类型进行输出。通过调用实现定义好的 result2.set（）来将结果定位到第二部分的显示界面上。并通过 result.set（a[:15]）截取字符串，来显示所得结果的长度。Btn 不仅能获取到用户输入的数字和运算符，还能接收到“退格”等相关操作的符号并存入 content 中。

用户输入



六、调试与测试

在课设过程中，我才用的调试方法是：遇到问题一步一步注释，确定问题来源，针对问题来源进行分析与修改。在刚开始的测试过程中，遇到的主要问题是：在运算 \cos 或 \sin 等相对复杂的运算时，会出现计算不准的原因，在多家分析之后发现如果用户直接输入“ \cos （”，并在结尾补上“）”的话就不会出现这类问题，所以，我将原来用户点击“ \cos ”输入 \cos 改成了输入“ \cos （”，

```
elif btn == 'sin':  
    content += 'sin('   
elif btn == 'cos':  
    content += 'cos('   
elif btn == 'log':
```

```

content += 'log('
elif btn == '√':
content += 'sqrt('

```

出现的这类带括号问题，经过我们分析加上左侧括号之后就解决了。

视图例子：

至此，课设项目运行所出现的问题都解决了。

七、设计总结

在徐老师辛勤教导的这十几周时间里，我对 Python 编程充满了兴趣，并通过老师所教授的只是完成了本次课设。本次课设过程中，我最深的体会是：编程性质的课程，如果不手动操作值是单纯学习书本上的知识是不能掌握的，在平常的学习中，我感觉都已经听懂了老师课上所讲，但是到自己编程的时候总是忘东往西，丢三落四的导致代码不完成，程序无法毫无 bug 的运行，通过这次课设，我对 Python 的基础语法的相关函数的使用有了深刻的了解，并能独立完成程序的运行。在本次课设中，存在着我们未完成的一下问题，就比如说：如过不将用户输入的“cos, sin, log”等相关运算符转换成“cos(, sin(, log(”，那么运算就会出现一些问题致使结果不准确。这个问题我们分析之后发现，是因为

eval 并不是 Python 中专门用于运算的函数，我们需要引入其他的专业用于计算的 python 库来对这个问题进行优化。最后，十分感谢学问碰老师这些天的辛勤教导，是您的孜孜不倦谆谆善诱让我了解到编程是一件另人愉悦的事，课设的项目成功运行，给了我们极大的成就感。

八、参考文献

- [1] 车万翔 译 《Python 程序设计》机械工业出版社. 2009
- [2] 张若愚 《Python 科学计算》 清华大学出版社. 2012
- [2] 陈志文等 译 《Python 与 tkinter 编程》 国防工业出版社. 2002

九、源程序

```
from tkinter import *
from math import *
root = Tk()
root.title('简易计算器')
root.geometry('350x770')
root.resizable(width=False,height=False)
l = Label(root, text="简易计算器", bg="white", fg='black', font=("Arabic",
20), width=400, height=3)
l.pack(side=TOP)
result = StringVar()
result.set('0')
result2 = StringVar()
result2.set('')
label = Label(root, font = ('微软雅黑', 20), bg = '#EEE9E9', bd = '9', fg =
'#828282', textvariable = result2)
label.place(x=0, y=80, width = 350, height = 200)
label2 = Label(root, font = ('微软雅黑', 35), bg = '#EEE9E9', bd = '9', fg =
'black', textvariable = result)
label2.place(x=0, y=200, width = 400, height =150)
#按钮功能设置
def buttonClick(btn):
    content = result2.get()
    a='',
```

```

if btn in '()e%123+456-789*.0/,':
    content += btn
elif btn in '←':
    content =content[:-1]
elif btn == 'π':
    content += 'pi'
elif btn == 'sin':
    content += 'sin('
elif btn == 'cos':
    content += 'cos('
elif btn == 'AC':
    content = ''
elif btn == '^2':
    content += '**2'
elif btn == 'log':
    content += 'log('
elif btn == '√':
    content += 'sqrt('
elif btn == '=':
    a = '='+str(eval(content))
result2.set(content)
result.set(a[:15])
#16 个符号
number
=['AC','(',')','e','←','^2','√','sin','cos','%','1','2','3','+','log',
'4','5','6','-',' ','7','8','9','*',' ','.','0','π','/','']
index = 0
for row in range(6):
    for col in range(5):
        num = number[index]
        index+=1
        btnDight = Button(root,text=num,font = ('微软雅黑',20),fg =
('black'),command=lambda x=num:buttonClick(x))
        btnDight.place(x=col*70,y=350+row*70,width=70,height=70)
# 等号美化

```

```

btnDight = Button(root, text='', font = ('微软雅黑', 40), fg =
('black'), bg=('gold'), command=lambda x='':buttonClick(x))
btnDight.place(x=280, y=330+300, width=70, height=140)
# AC美化
btn = Button(root, text='AC', font = ('微软雅黑', 20), fg =
('Black'), bg=('red'), command=lambda x='AC':buttonClick(x))
btn.place(x=0, y=350, width=70, height=70)

root.mainloop()

```