

**《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 马志豪

专业班级 信管1601

学生学号 311609030118

教师姓名 徐 文 鹏

成 绩**：**

评 语：

教师签名：

日期：

[一、 设计目的 - 3 -](#_Toc3881)

[1.课程设计目的： - 3 -](#_Toc11062)

[二、 设计任务和要求 - 3 -](#_Toc30278)

[1.设计任务 - 3 -](#_Toc13096)

[2.设计要求 - 3 -](#_Toc31782)

[3.设计效果图 - 4 -](#_Toc10863)

[三、 总体设计 - 4 -](#_Toc11065)

[1.负责分工： - 5 -](#_Toc20375)

[2.功能实现： - 5 -](#_Toc20794)

[1)加减乘除括号运算 - 5 -](#_Toc5964)

[2)其他运算 - 5 -](#_Toc24212)

[3)程序流程图 - 6 -](#_Toc17166)

[四、复杂计算功能详细设计 - 7 -](#_Toc11065)

[五、调试与测试 - 8 -](#_Toc15698)

[六、 设计总结 - 9 -](#_Toc15692)

[七、 参考文献 - 10 -](#_Toc4643)

[八、 源程序 - 11 -](#_Toc12779)

**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

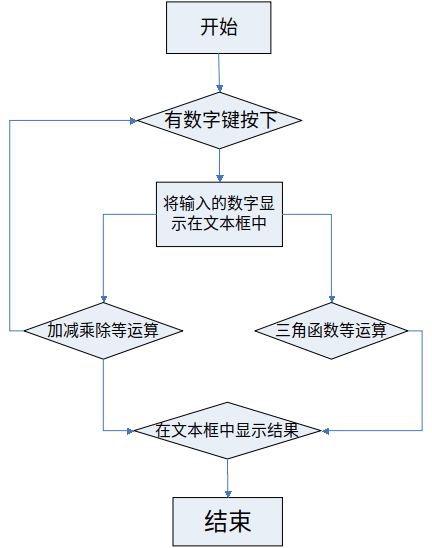
2. 总体上基于python3+，python GUI Tkinter语法编程

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)

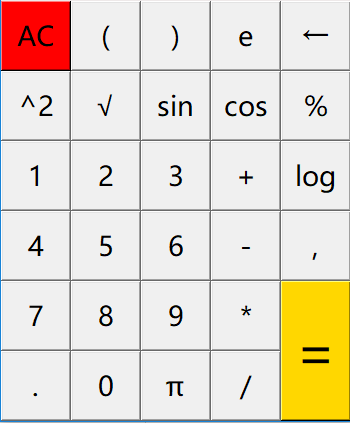
这里用到了math来帮助我们求得运算结果，我们统一交给eval处理。

**三、复杂计算功能详细设计**

1，现在需要在原有功能上增加一些复杂的运算，包括三角函数的计算，对数运算，平方运算，开跟运算，并增加圆周率的键位，使用这些函数，首先要引入python的math库，math库中为我们提供了这些操作对应的函数。

2，首先我们要找到程序中的图形化的按钮部分，将我们需要添加的功能的键位补充。可以看出，number变量中记录按键的显示，并用number中的位置表示按键的位置，我们将这些按键添加上去。代码：

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

调整下方遍历的次数。由5\*4变为6\*5，最终样式为

3，然后进行按钮功能的实现。根据现有的函数流程可以看出，采用的是比较简单的方式，即将需要计算的式子交给python，python对这段代码进行运算。所以在原有代码上，增加ifelse 语句，可以完成这些函数的运算。代码：

elif btn == 'sin':

content += 'sin('

elif btn == 'cos':

content += 'cos('

elif btn == '^2':

content += '\*\*2'

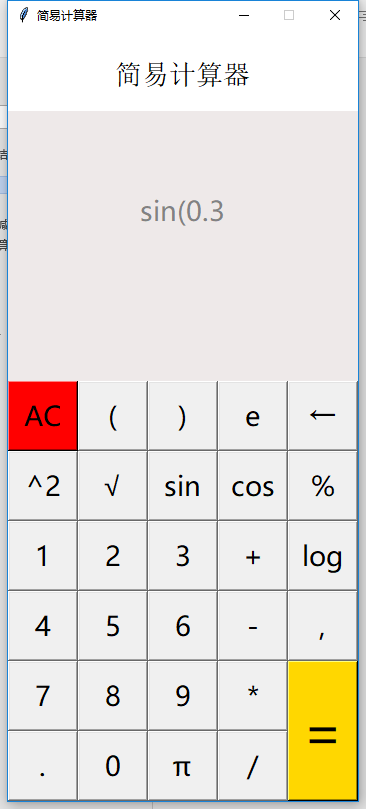
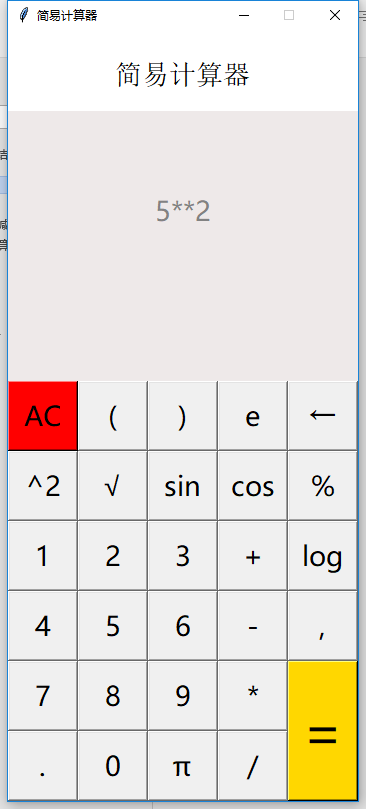
elif btn == 'log':

content +='log('

elif btn == '√':

content +='sqrt('

然后最后运用eval(content)进行计算，但是需要注意的是，因为直接嵌套在ifelse语句中，所以既不能不写括号，也不能将括号写完整，所以需要使用半括号，另一半括号由用户自己补充，最终显示的结果如下



4，流程图

Btn=

=’Sin(‘

content += 'sin('

content += 'cos('

=’cos(‘

content += '\*\*2'

=’^2’

v

content +='log('

=’sqrt(‘

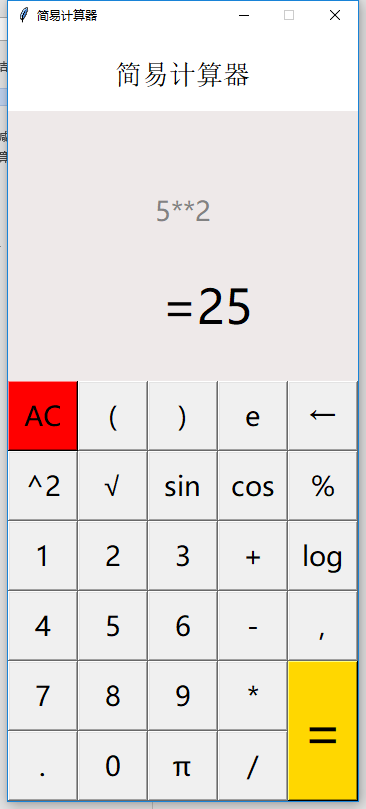
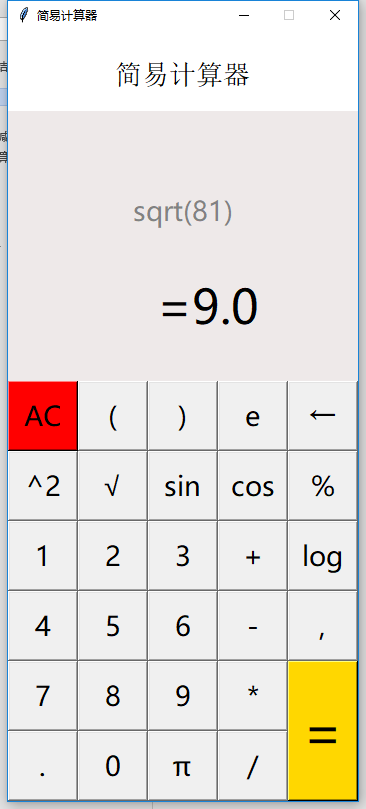
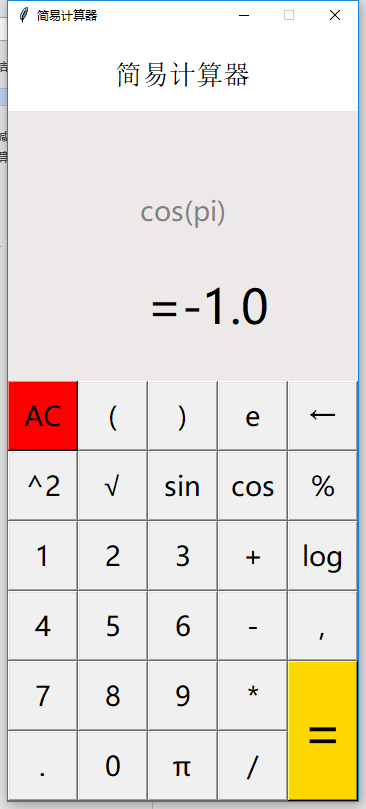
result2.set(content)

result.set(a[:15])

**四、调试与测试**

**1，测试平方根，开跟，对数的计算**

属于一些数值，进行计算，结果均正确

**2，测试三角函数**

测试三角函数。Cos结果正确，sin结果不正确。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。

这时需要我们对它进行简单的处理，当匹配到sin时，sin（π/2）可以正常计算，但是sin（π）无法正常计算 ，所以需要最程序最后判断一下,如果python计算的此段程序中有sin（π），则直接替换为1。代码如下：

if content == 'sin(pi)':

content = 1

**3，连续测试，计算多个式子的开根取余，也可以正确计算**

在测试时，功能正常运算，但是发现需要为函数补充括号，否则运算就不能正常进行。

**六、设计体会与小结**

在本学期我们学习了python课程，并在这段时间里完成了此次的课程设计。在进行课程设计过程中，难免会遇见一些困难，当然也学习到了很多东西。其中遇到的困难有：1，前期对项目的理解与分工不同。在确定项目选题后，就要进一步考虑项目分工，就此次课程设计而言，计算形式成为讨论的点，对于使用python语句直接计算，和使用函数进行计算产生分歧；最后决定由负责计算部分的人敲定，考虑到时间问题，和此计算器不需要太严格的要求，所以采用了这种简单的形式，可以快速开发，会大大减少开发成本。事实证明，因为团队成员都懂得使用git，使开发效率十分快，使我们一上午时间就完成程序的第一版。2，技术问题，因为小组成员对python并不是特别擅长，所以有些问题还是需要百度或者询问其他同学，比如sin(π)的问题，查阅资料之后才知道是精度导致的计算问题，然后去解决。最后设计完成之后也有些不足之处，比如这些sin括号不能不全，π显示为pi，等影响了计算机的使用，需要进一步完善和调整，最后肯定要封装为特定的函数。在学习python过程中，在进行课程设计过程中也有很多收获，其中最主要的还是对python这门语言有了更加深入的了解，因为接触过其他语言，所以在使用python上初期觉得与其他语言很不“像”，循环或者字符串的一些操作都十分特别，但是在完成一些练习和课程设计后，确实发现，在完成相同功能的使用的语句更加的少，虽然python这门课程结束了，但是在python火热的时代形势下，以后接触python的机会更多，需要使用python的地方也会更多，也希望能够成为其中的一员，在程序设计的道路上越走越远，敢于创新，改变世界。

**七、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

**八、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))

if content == 'sin(pi)':

content = 1

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()