

**《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 程龙

专业班级 信管1601

学生学号 311609030114

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： - 4 -](#_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、图形化界面详细设计 - 7 -](#_Toc11065)

[五、设计结果与分析 - 8 -](#_Toc15698)

1. [运行调试 - 8 -](#_Toc2255)
2. [结果分析 - 8 -](#_Toc2255)

[六、 设计体会与小结 - 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**

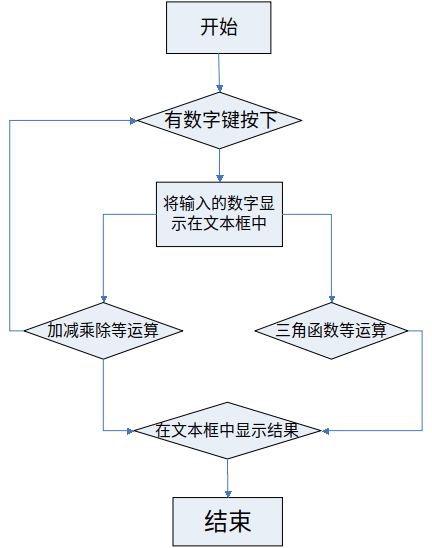
对于三角函数运算，开根，平方，对数运算，主要用到了Python内置的math函数库，通过eval自动处理函数运算结果实现。例如求sin pi。则是

import math

math.sin(math.pi)

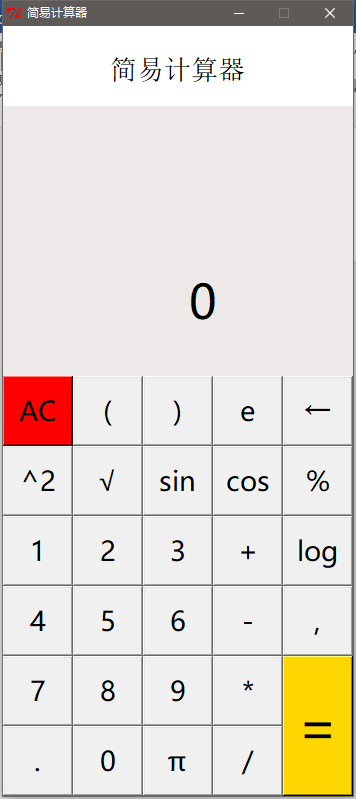
这里用到了math来帮助我们求得运算结果，我们统一交给eval处理。

**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**

**⒈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**

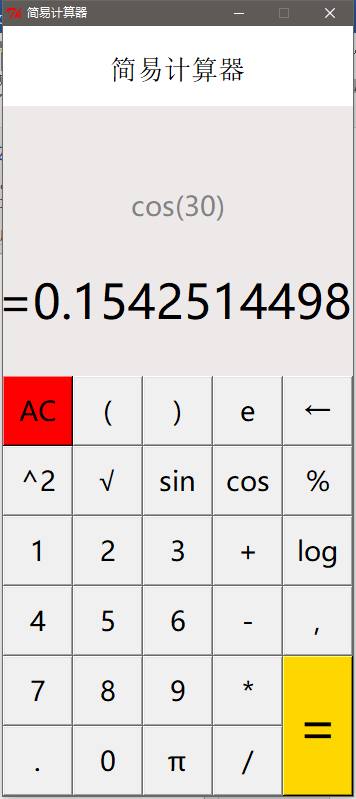
按下AC键清零

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。

**六、设计体会与小结**

作为一名计算机科学与技术专业的学生，我觉得能做类似的课程设计是十分有意义，而且是十分必要的。经过一段时间的努力，计算器课程设计终于结束了，虽然很忙碌、很疲劳，但是收获很大。这些天我几乎每天的专注和辛劳，唤回了我对Python这门语言的重新的认识，对Python这门编程语言也有了深刻的理解。当然在本次课程设计的过程中，我碰到了很多问题，比如求倒时，最先我是把按钮新建成“1/x”，但是出现问题，怎么找都没找出问题来，其后重新整理思绪才发现不能设成“1/x”。还有就是实现退格功能时不知道怎样处理，最后上网百度了一下才找到了方法，这也让我深刻体会到了网络的作用。当然还遇到了其他的困难，这些花费了我很多时间，使我失败了很多次，但是通过我们不解的努力还是一一克服了这些问题，最终还是成功了。当然在这次试验中也暴露自己一些问题，发现了自己的很多不足，自己知识的很多漏洞，基础知识的不牢固，看到了自己的实践经验还是比较缺乏，理论联系实际的能力还急需提高。

这次课程设计，除了全面系统的理解了Python面向对象编程原理的一般原理和基本实现方法，还把死板的课本知识变得生动有趣，激发了学习的积极性。把学过的编程原理的知识强化，把课堂上学的知识通过自己设计的程序表示出来，这加深了对理论知识的理解。课程设计中程序比较复杂，在调试时应该仔细，在程序调试时，注意尽力寻找程序中存在的不足之处，进而使功能更加完善。当然在做课程设计要有很大的耐心，做的过程就是不断的调试—修改—调试过程，只有静下心来才能把问题解决。

**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))

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()