制作一个计算器

一、项目介绍

使用tkinter库，制作一个计算器，实现简单的加、减、乘、除、平方、倒数的运算。tkiner库是Python内置库，以下代码直接复制到本地编辑器上运行即可。

二、项目代码与结果展示

2.1 项目代码

```
from tkinter import Tk,Button,Text,END
import math
# 定义一个栈
class Calculator():
    def __init__(self):
        self.items = []
    def isEmpty(self):
        return self.items == []
    def push(self,item):
        self.items.append(item)
    def pop(self):
        return self.items.pop()
    def peek(self):
        return self.items[-1]
    def size(self):
        return len(self.items)

# 将计算式转化为后序表达式
def postfixExpr(formulaList):
    # 定义等级
    prior = {}
    prior["*"] = 3
    prior["/"] = 3
    prior["+"] = 2
    prior["-"] = 2
    prior["("] = 1
```

```
27
28      opstack = Calculator()
29      postfixList = []
30      for token in formulaList:
31          if token =='(':
32              opstack.push(token)
33          elif token == ")":
34              topToken = opstack.pop()
35              while topToken != "(":
36                  postfixList.append(topToken)
37                  topToken = opstack.pop()
38          elif token in "*/+-":
39              while (not opstack.isEmpty()) and (prior[opstack.peek()]>= prior[token]) :
40                  postfixList.append(opstack.pop())
41              opstack.push(token)
42          else :
43              postfixList.append(token)
44
45      while not opstack.isEmpty():
46          postfixList.append(opstack.pop())
47
48      return " ".join(postfixList)    # 返回后序表达式
49
50  # 后序表达式计算
51  def postfixEval(postfixExpr):
52      operandStack = Calculator()
53      tokenLIst = postfixExpr.split()
54      for token in tokenLIst:
55          if token in "*/+-":
56              number1 = operandStack.pop()
57              number2 = operandStack.pop()
58              result = doMath(token, number1, number2)
59              operandStack.push(result)
60          else:
61              operandStack.push(float(token))
62      return operandStack.pop()  # 返回计算结果
63
64  # 分步计算函数,
65  def doMath(op,num1,num2):
66      if op == "*":
67          return num2 * num1
68      elif op == "/":
69          return num2 / num1
70      elif op == "+":
```

```
        return num2 + num1
    elif op == "-":
        return num2 - num1

# 定义Value类，用于把值传到GUI界面的Text框中
class Value():

    def __init__(self,value):
        self.value = value
    # 将输入按钮的值传到文本框中
    def insert_value(self):
        self.result_text=Application.result_text
        self.result_text.insert(END, self.value)

# 定义Application类，用于画GUI和计算结果。
class Application(object):
    window = Tk()   # 创建一个GUI界面
    # 创建一个文本框
    result_text = Text(window, background='azure')
    # 更多背景色：http://www.science.smith.edu/dftwiki/index.php/Color_Charts_for_TKinter
    result_text.place(x=12.5, y=8, width=325, height=60)

    def __init__(self):
        self.window.title(u'我的计算器') # 定义界面名称

        # 设置窗口大小和位置
        self.window.geometry('350x310+500+300')
        self.window.minsize(350,315)
        self.window.maxsize(350,315)

        # 第1行按钮
        self.submit_btn0 = Button(self.window,text=u'(',command=Value('(').insert_value)
        self.submit_btn0.place(x=12.5,y=76,width=60,height=40)
        self.submit_btn1 = Button(self.window,text=u'C',command = self.clean)
        self.submit_btn1.place(x=78.75,y=76,width=60,height=40)
        self.submit_btn2 = Button(self.window, text=u'÷', command=Value('/').insert_value)
        self.submit_btn2.place(x=145, y=76, width=60, height=40)
        self.submit_btn3 = Button(self.window, text=u'×', command=Value('*').insert_value)
        self.submit_btn3.place(x=211.25, y=76, width=60, height=40)   # 12.5+(60+6.25)*3
        self.submit_btn4 = Button(self.window, text=u'←', command=self.backspace)
        self.submit_btn4.place(x=277.5, y=76, width=60, height=40)
        # 第2行按钮
        self.submit_btn5 = Button(self.window,text=u')',command=Value(')').insert_value)
        self.submit_btn5.place(x=12.5,y=76+8+40,width=60,height=40)
```

```
        self.submit_btn6 = Button(self.window,text=u'7',command = Value(7).insert_value)
        self.submit_btn6.place(x=78.75,y=124,width=60,height=40)
        self.submit_btn7 = Button(self.window, text=u'8', command=Value(8).insert_value)
        self.submit_btn7.place(x=145, y=124, width=60, height=40)
        self.submit_btn8 = Button(self.window, text=u'9', command=Value(9).insert_value)
        self.submit_btn8.place(x=211.25, y=124, width=60, height=40)
        self.submit_btn9 = Button(self.window, text=u'-', command=Value('-').insert_value)
        self.submit_btn9.place(x=277.5, y=124, width=60, height=40)
        # 第3行按钮
        self.submit_btn10 = Button(self.window,text=u'π',command=Value(math.pi).insert_value)
        self.submit_btn10.place(x=12.5,y=172,width=60,height=40)
        self.submit_btn11 = Button(self.window,text=u'4',command = Value(4).insert_value)
        self.submit_btn11.place(x=78.75,y=172,width=60,height=40)
        self.submit_btn12 = Button(self.window, text=u'5', command=Value(5).insert_value)
        self.submit_btn12.place(x=145, y=172, width=60, height=40)
        self.submit_btn13 = Button(self.window, text=u'6', command=Value(6).insert_value)
        self.submit_btn13.place(x=211.25, y=172, width=60, height=40)
        self.submit_btn14 = Button(self.window, text=u'+', command=Value('+').insert_value)
        self.submit_btn14.place(x=277.5, y=172, width=60, height=40)
        # 第4行按钮
        self.submit_btn15 = Button(self.window,text=u'1/x',command= Value('1/').insert_value)
        self.submit_btn15.place(x=12.5,y=220,width=60,height=40)
        self.submit_btn16 = Button(self.window,text=u'1',command = Value(1).insert_value)
        self.submit_btn16.place(x=78.75,y=220,width=60,height=40)
        self.submit_btn17 = Button(self.window, text=u'2', command=Value(2).insert_value)
        self.submit_btn17.place(x=145, y=220, width=60, height=40)
        self.submit_btn18 = Button(self.window, text=u'3', command=Value(3).insert_value)
        self.submit_btn18.place(x=211.25, y=220, width=60, height=40)
        self.submit_btn19 = Button(self.window, text=u'=', command=self.equal)
        self.submit_btn19.place(x=277.5, y=220, width=60, height=88)
        # 第5行按钮
        self.submit_btn20 = Button(self.window,text=u'x²',command=Value('^2').insert_value)
        self.submit_btn20.place(x=12.5,y=268,width=60,height=40)
        self.submit_btn21 = Button(self.window,text=u'%',command = Value('%').insert_value)
        self.submit_btn21.place(x=78.75,y=268,width=60,height=40)
        self.submit_btn22 = Button(self.window, text=u'0', command=Value(0).insert_value)
        self.submit_btn22.place(x=145, y=268, width=60, height=40)
        self.submit_btn23 = Button(self.window, text=u'.', command=Value('.').insert_value)
        self.submit_btn23.place(x=211.25, y=268, width=60, height=40)

    def clean(self):        # 清空键
        self.result_text.delete(0.0, END)
        return
    def backspace(self):    # 退格键
```

```
        content = self.result_text.get(0.0, END).strip().split("\n")
        self.result_text.delete(0.0, END)
        for i in content[:-1]:
            self.result_text.insert(END, i)
            self.result_text.insert(END, '\n')

        self.result_text.insert(END, content[-1][:-1])
        self.result_text.see(END)  # 定位光标到最后

    def equal(self):        # 等于键
        formula = self.result_text.get(0.0, END).strip().split("\n")[-1]  # 获取界面的计算式子
        print(formula, end=" ")
        formula = formula.replace("^2", "^")  # 将平方保留一个符号^方便后面计算
        try:
            # 将数字和运算符号分离
            formulaList1 = []
            i = 0
            while i < len(formula):
                if formula[i] in "0123456789.":
                    cc = ''
                    while i < len(formula):
                        if formula[i] in "0123456789.":
                            cc += formula[i]
                            i += 1
                        else:
                            break
                    i -= 1
                    formulaList1.append(cc)
                else:
                    formulaList1.append(formula[i])
                i += 1
            # 将平方和百分号做处理
            formulaList2 = []
            for i in formulaList1:
                formulaList2.append(i)
                if i == "^":
                    index = formulaList1.index("^")
                    formulaList1[index] = "-"   # 将^符号替换掉，方便查找第2个^
                    number = formulaList1[index - 1]
                    formulaList2.pop()
                    formulaList2.append('*')
                    formulaList2.append(number) # 转化为两个数相乘
                elif i == "%":
                    index = formulaList1.index("%")
```

```
                    formulaList1[index] = "-"    # 将%符号替换掉，方便查找第2个%
                    formulaList2.pop()
                    formulaList2[index-1]=str(float(formulaList2[-1])/100)  # 转化为小数

            global result    # 方便在except中调用
            b = postfixExpr(formulaList2)       # 获取后序表达式
            result = postfixEval(b)  # 获取计算结果值
            print(result)
            self.result_text.insert(END, '=')
            self.result_text.insert(END, result)
            self.result_text.insert(END, '\n')  # 另起一行
            self.result_text.see(END)   # 定位光标到最后

        except ValueError:
            self.result_text.insert(END, result)
            self.result_text.insert(END, '\n')
            self.result_text.see(END)  # 定位光标到最后
            print(result)
        except:
            self.result_text.insert(END, 'Error!')
            self.result_text.insert(END, '\n')
            self.result_text.see(END)   # 定位光标到最后
            print("Error!")

    def run(self):  # 运行tkinter，展示界面
        self.window.mainloop()

if __name__=="__main__":
    app = Application()
    app.run()
```


2.2 结果展示

直接运行显示结果：

运算显示结果：