**计算器设计分析报告**

**软件设计题目：计算器**

**设计内容：计算器简单的加减乘除功能**

**组内成员：王杰 王宏基 王泽普 文光顺**

**设计人员：文光顺**

**一．设计简介**

**1.1设计目的**

21世纪计算机的发展突飞猛进，人们使用计算机的目的各不相同，但是我们很容易发现，关于计算的问题生活中随处可见，于是计算器也成为了我们学习和生活中必不可少的工具。为了减轻人们日益繁琐的工作压力和负担，实现高效统计和运算，所以使用计算器势在必行，可以大大减轻会计和财务部门统计账目的压力，于是计算器应运而生。

**1.2 设计内容及要求**

请用 Python3 或 Java(JDK1.8)编写一个计算器的控制台程序，支持加减乘除、乘方、括号、小 数点，运算符优先级为括号>乘方>乘除>加减，同级别运算按照从左向右的顺序计算。

**1.3输入描述**

数字包括"0123456789"，小数点为"."，运算符包括：加("+")、减("-")、乘("\*")、除("/")、乘方 ("^"，注：不是\*\*！)、括号("()") 需要从命令行参数读入输入，例如提交文件为 main.py，可以用 python3 main.py "1+2-3+4"的 方式进行调用，Java 程序也是类似的，如果你的程序需要通过键盘输入，那么是不符合要求 的，例如 python 使用 input()来等待用户输入，这会因为自动评测时不会有用户输入所以不 会有任何结果。 输入需要支持空格，即 python3 main.py "1 + 2 - 3 + 4" 也需要 程序能够正确给出结果，Java 程序也是类似的 所有测试用例中参与运算的非零运算数的绝对值范围保证在 10^9-10^(-10) 之内, 应该输出 运算结果时非零运算结果绝对值也保证在该范围内

**1.4输出描述**

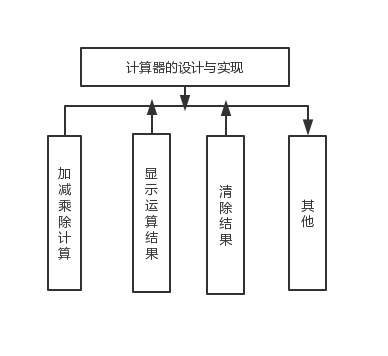
数字需要支持小数点，输出结果取 10 位有效数字，有效数字位数不足时不能补 0 对于不在输入描述内的输入，输出 INPUT ERROR 对于格式不合法（例如括号不匹配等）的输入，输出 FORMAT ERROR 对于不符合运算符接收的参数范围（例如除 0 等）的输入，输出 VALUE ERROR 对于 2、3、4 的情况，输出即可，不能抛出异常 同时满足 2、3、4 中多个条件时，以序号小的为准

**二．概要设计**

**2.1，设计思路**

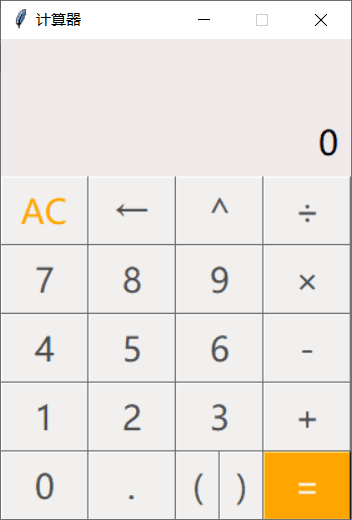
此次课程设计实践，主要可以实现四则运算等，并用图形界面实现。可以培养我们严谨务实的工作态度以及提高我们分析问题和解决问题的能力，也对于我们对面向对象程序设计的基本概念有所复习，基本理论和实现技术的理解有起到了一定的意义。

**2.2功能模块图**

****

**三．详细设计**

**3.1.界面设计效果图展示**



**3.2 计算模块函数代码展示**

#设置参数

lists = [] #设置一个变量，保存运算数字和符号的列表

isPressSign = False #添加一个判断是否按下运算符的标志,假设默认没有按下按钮

isPressNum = False #添加一个判断是否按下数字的标志,假设默认没有按下按钮

#数字函数

def pressNum(num): #设置一个数字函数，判断是否按下数字，并获取数字将数字写在结果显示面板上

global lists #全局化lists和按钮状态isPressSign

global isPressSign

if isPressSign == False:

pass

else: #重新将运算符号状态设置为否

result.set(0)

isPressSign = False

#判断界面的数字是否为0

oldnum = result.get() #获取数字

if oldnum =='0': #如果界面上的数字是0，则获取按下的数字

result.set(num)

else: #如果界面上的数字不是0，则链接上新按下的数字

newnum = oldnum + num

result.set(newnum) #将按下的数字写到面板中

#运算函数

def pressCompute(sign):

global lists

global isPressSign

lists.append(result.get()) # 保存界面数字到列表中

result.set('')

lists.append(sign) # 将按下的运算符号保存到列表中

isPressSign = True

if sign =='AC': #如果按下的是‘AC’按键，则清空列表内容，讲屏幕上的数字键设置为默认数字0

result.set(0)

result2.set('')

lists.clear()

if sign =='B': #如果按下的是退格‘Backspace’，则选取当前数字第一位到倒数第二位

result.set(str(result.get()[:-1]))

#获取运算结果函数

def pressEqual():

global lists

global isPressSign

try:

curnum = result.get() #设置当前数字变量，并获取添加到列表

lists.append(curnum)

computrStr = ''.join(lists) #将列表内容用join命令将字符串链接起来

endNum = eval(computrStr) #用eval命令运算字符串中的内容

a = str(endNum)

# b = '='+a #在运算结果前添加一个‘=’，但是这样会出现BUG，不能进行连续运算，所以这里注释掉，不要‘=’

c = a[0:12] #所有的运算结果取10位数

result.set(c) #讲运算结果显示到屏幕1

result2.set(computrStr) #将运算过程显示到屏幕2

lists.clear() #清空列表内容

except ZeroDivisionError: #除零异常

result2.set(lists)

result.set('VALUE ERROR')

except SyntaxError: #格式异常

result2.set(lists)

result.set('FORMAT ERROR')

except IndexError: #参数异常

result2.set(lists)

result.set('INDEX ERROR')

except : #异常

result2.set(lists)

result.set('ERROR')

root.mainloop() #启动窗体运行，并等待接收各种事件信息

**四．设计心得**

此次计算器的设计让我更熟悉更了解Python中图形用户界面的编程方法。在完成这次课题的过程中也不断充实了自己，还学习到了很多以前没有学习到的知识，收获还算比较大。感觉在这次的完成过程中也培养了自己解决问题的能力，在学习中肯定会有很多的问题等着我们去解决，会有很多自己不会的东西，我们要努力找到解决问题的方法。计算器的设计使得我们对所学的知识能够融会贯通，又不断丰富了新知识。