说明文档

所用包部分

```
from tkinter import Tk, StringVar, Label, Button, Entry, Radiobutton
from tkinter.messagebox import showinfo
```

`tkinter`：Tkinter 模块是 Python 的标准 `Tk GUI` 工具包的接口。`Tk` 和 `Tkinter` 可以在大多数的 Unix 平台下使用,同样可以应用在 `Windows` 和 `Macintosh` 系统里。`Tk8.0` 的后续版本可以实现本地窗口风格,并良好地运行在绝大多数平台中。

```
Tk #创建主窗口.
StringVar   # 使用界面编程的时，跟踪变量的值的变化，以保证值的变更随时可以显示在界面上(黄色背景上的文字).
Label   # 在 Tkinter 中，Label 控件用以显示文字和图片. Label 通常被用来展示信息，而非与用户交互。这里与StringVar搭配，使得其可动态显示.
Button  # 一种标准 Tkinter 控件，用来展现不同样式的按钮. Button 控件被用以和用户交互.
Entry   # Entry 是 Tkinter 用来接收字符串等输入的控件.
Radiobutton # 选择按钮 A B C ......
showinfo    # 消息弹窗
```

```
from numpy import mean, var
```

`numpy`：`NumPy`是 `Python` 语言的一个扩展程序库，支持大量的维度数组与矩阵运算，此外也针对数组运算提供大量的数学函数库。

```
mean    # 返回数组中元素的算术平均值.
var     # 统计中的方差（样本方差）是每个样本值与全体样本值的平均数之差的平方值的平均数.
```

```
from math import sqrt, log10
```

`math`：Python math 模块提供了许多对浮点数的数学运算函数。

```
sqrt(x)     # 用来求 x 的平方根.
log10(x)    # 用来求以 10 为底 x 的对数.
```

部分总结：运用以上功能，进行逻辑拼接，便可完善所需功能。

具体逻辑部分(主要方法，次要方法源码中有注释，不难理解)

类属性

```
window = Tk()
window.title('Calculation window')  # 设置窗口的名称
window.geometry('500x300')  # 设置窗口大小尺寸
kw = 10 ** 14   # 定义Kw常量
num_a = 65  # 设置常数，用于 ASCII 与大写字母的转换
num_b = 71
num_c = 65
```

```
ef_load_judge = False    # 设置布尔类型的值
F = False    # 主要用于判断 F 选项是否被触发
number_arr = []      # 将输入框中的字符分割转化为数字类型的list
string_arr = []      # 储存输入框中的字符
cur_string_arr = []      # 暂储
cur_number_arr = []      # 暂储
var = StringVar()    # 所用包部分有讲述
```

定义相关类变量，使类更加丰满，利用其类变量，可以使程序处理起来更加灵活。

```
dicts_main = {
        'A': '平均值',
        'B': '平均偏差',
        'C': '相对平均偏差',
        'D': '标准偏差',
        'E': '相对标准偏差',
        'F': '溶液H+离子浓度',
        'G': '强酸',
        'H': '一元弱酸',
        'I': '两性物质',
        'J': '缓冲溶液'
    }

dicts_middle = {
    'A': '一位',
    'B': '两位',
    'C': '三位',
    'D': '四位'
}
```

借用 `Python` 语言的重要数据类型，储存选择项的值。

实例属性

```
self.lab = Label(self.window, bg='yellow', width=60, height=2,
text='Please check method.')  # 初始化Label 控件用以显示文字
self.lab.pack() # 使得Lable控件显示
self.btn = None # 初始化Button控件
self.entry = None # 初始化Entry控件
self.option = None # 初始化保存选择的选项实例属性 A B C D E
self.option2 = None # 初始化保存选择的选项实例属性 G H I J
self.sure_btn = None # 初始化Button控件
self.ef_digits = None # 初始化保留位数储存的实例属性
self.reset_btn = None # 初始化Button控件
```

类静态方法部分

```
@staticmethod   # A B C D E F后跟的是lambda表达式，也称匿名函数，搭配字典的键值对取值，极大简化了代码，lambda:前方为参数列表，后方为具体算法逻辑.
def switch(item):
    switcher = {
        "A": lambda res: mean(res),
        "B": lambda res: mean(list(map(lambda x: abs(x - mean(res)), res))),
        "C": lambda res: mean(list(map(lambda x: abs(x - mean(res)), res))) / mean(res),
        "D": lambda res: sqrt(var(res)),
        "E": lambda res: sqrt(var(res)) / mean(res),
    }
    return switcher.get(item, 'No that method.')   # 键值对取法，例如提供参数item = A，返回A后的lambda匿名函数
```

由于`Python`语言中没有`switch`语法，所以利用其重要数据类型及其操作方法 -- 字典来进行自定义`switch`语法结构。

```
def switch2(self, item):
    switcher = {
        "G": lambda res: (res[0] + sqrt(res[0] ** 2 + 4 * self.__class__.kw)) / 2,
        "H": lambda res: sqrt(res[0] * res[1]),
        "J": lambda res: (-log10(res[0])) + log10(res[1] / res[2]),
    }
    return switcher.get(item, 'No that method.')
```

与`switch`相同，只不过不是静态方法，属于实例方法范畴。

实例方法部分

```
def pk_forget(self, name, num1=65, num2=70, judge=False): # name参数为要隐藏或显示的选择实例属性名，num1 num2  65 ASCII 值对应大写字母 A，以此类推
    for i in range(num1, num2 + 1):  # 为何是 num2+1，因为for range得循环截至到num2+1的前一个，所以只循环了num1-num2
        attr_name = name + chr(i) # chr(i)根据每次循环的值转换为大写字母
        if not judge:  # 根据judge参数 确定隐藏与否 默认参数False 隐藏控件
            getattr(self, attr_name).pack_forget() # # getattr 函数获取实例化对象中的属性(界面控件)
        else:
            getattr(self, attr_name).pack()
```

对于界面控件的显示与否的功能封装，用于简化代码。 由于在点击按钮时要将之前的选项给隐藏，而按钮是有规律的，若不封装，代码太过冗余。

```
def load_r(self, name, num1=65, num2=70, judge=True):
    for i in range(num1, num2 + 1):
        attr_name = name + chr(i)
        if num1 == 71: # 判断num1的值，要初始化的为G H I J
            self.__class__.F = True
            stt = self.__class__.dicts_main.get(chr(i))
            setattr(self, attr_name, Radiobutton( # setattr函数设置实例化对象中的属性(界面控件)
                self.window, text=chr(i) + ' ' + stt,
                variable=self.__class__.var, value=chr(i),
```

```
                command=self.print_ser1
            ))
            self.pk_forget('r1', self.__class__.num_b,
self.__class__.num_b, True)
            self.__class__.num_b = self.__class__.num_b + 1
        else:
            if judge: # 如果不是F，打印出选项，并在self.print_selection方法
里进入到保留位数的选项
                stt = self.__class__.dicts_main.get(chr(i))
                if chr(i) != 'F':
                    setattr(self, attr_name, Radiobutton(
                        self.window, text=chr(i) + ' ' + stt,
                        variable=self.__class__.var, value=chr(i),
                        command=self.print_selection
                    ))
                else: # 如果选了 F
                    setattr(self, attr_name, Radiobutton(
                        self.window, text=chr(i) + ' ' + stt,
                        variable=self.__class__.var, value=chr(i),
                        command=self.select_h
                    ))
                self.pk_forget('r', self.__class__.num_a,
self.__class__.num_a, True)
                self.__class__.num_a = self.__class__.num_a + 1
            else: # 初始化保留位数的选项组件
                stt = self.__class__.dicts_middle.get(chr(i))
                setattr(self, attr_name, Radiobutton(
                    self.window, text=chr(i) + ' ' + stt,
                    variable=self.__class__.var, value=str(i),
                    command=self.print_ser0
                ))
                self.pk_forget('r0', self.__class__.num_c,
self.__class__.num_c, True)
                self.__class__.num_c = self.__class__.num_c + 1
```

关于这一部分，大体内容为：

```
"""
初始化选项函数
三合一：
1、A B C D E F   变量为rA rB...
2、G H I J   变量为r1G r1H...
3、A B C D   变量为r0A r0B...(保留有效数字选项)
"""
# 关于这部分我不再每行解释了，此函数算是很重要的一个函数，只需注意到命名规范，便不难
看懂具体逻辑
# A B C D E F 此为页面最先加载的5个选项，命名规范为 rA rB ......且在字典中键是字
母，值是字符串，所以在第一个字典中将它们放在了一起，并根据所给参数num1 num2值得不同
来决定初始化ABC...(ASCII码65-70)还是GHI...(71-74)
# A B C D 关于保留位数 键为A B...，但是初始化的选项组件的value值为数字，所以，额
外分出来一类
# 注意str(i) 与 chr(i)的区别，str()仅仅将循环的i值转换成字符类型，chr(i)是将i值
与大写字母对应，依照ASCII码
# 注意 A B C D E选项控件所调用的方法与 F控件有所不同
# 前者 command=self.print_selection  后者 command=self.select_h
```