**计算器-软件测试过程文档**

根据源程序代码可以知：

Calculator计算器类

包含了三个函数，

其中一个是初始化函数，一个是程序事件处理函数，一个程序界面函数。

其中对于计算器的关键功能点”计算”起作用的函数是程序事件处理函数ButtonClick，

故此次软件测试，将主重测试ButtonClick函数，但因 ButtonClick函数需要传递进入鼠标点击事件，故用测试用例测试该函数。

from tkinter import \*  
Str = ('~','|','&','↺','1', '2', '3', '+', '4', '5', '6', '-', '7', '8', '9', '\*', '.', '0', '/', '=')  
  
class Calculator:  
 def \_\_init\_\_(self, master):  
 self.master = master  
 self.Interface()  
  
#程序事件处理  
 def ButtonClick(self,var):  
 # 获取文本框中的内容  
 textcontent = self.content.get()  
 # 判断点击消息，将textcontent设置为要计算的式子，并在文本框显示  
 if var.widget['text'] in '0123456789.+-\*/&|~':  
 textcontent += var.widget['text']  
 self.content.set(textcontent)  
 # 使用eval计算textcontent，并在文本框显示  
 elif var.widget['text'] == '=':  
 self.content.set(eval(textcontent))  
 # 清除文本框的内容  
 elif var.widget['text'] == '↺':  
 textcontent = ''  
 self.content.set(textcontent)  
  
# 程序界面接口  
 def Interface(self):  
 # 设置文本框显示的字符串  
 self.content = StringVar(self.master, '')  
 # 创建Entry组件  
 self.module = Entry(relief=SUNKEN, font=('Arial',25), width=26,textvariable=self.content)  
 self.module.pack(side=TOP, pady=10)  
 # Grad布局  
 fra = Frame(self.master)  
 fra.pack(side=TOP)  
 # 创建Button组件  
 for i in range(len(Str)):  
 botton = Button(fra,text=Str[i], font=('Arial',25),width=6)  
 botton.grid(row=i//4,column=i%4)  
 botton.bind('<Button-1>',self.ButtonClick)  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 root = Tk()  
 root.title('计算器')  
 Calculator(root)  
 root.mainloop()

该函数中

self.content.set(eval(textcontent))

此行代码代码起到了计算作用

测试用例包含此计算器可计算的加减乘除:

测试用例1：1+2+…+99+100=5050

测试用例2：-1-2-…-99-100=-5050

测试用例3：

1\*2\*…\*99\*100=93326215443944152681699238856266700490715968264381621468592963895217599993229915608941463976156518286253697920827223758251185210916864000000000000000000000000

测试用例4:1/2/…/9/100=1.071510288125467e-158

测试用例5：1+2-3+4-5…+98-99+100=52

测试用例6:1\*2/3\*4/…\*98/99\*100=12.564512901854913

测试用例：1+2-3\*4/5=0.6000000000000001

测试代码：

import pytest

def Test\_eval(textcontent):  
 return eval(textcontent)  
  
def test\_answer():  
 assert Test\_eval("1+2+3+4+5+6+7+8+9+10+11+12+13+14+15+16+17+18+19+20+21+22+23+24+25+26+27+28+29+30+31+32+33+34+35"  
 "+36+37+38+39+40+41+42+43+44+45+46+47+48+49+50+51+52+53+54+55+56+57+58+59+60+61+62+63+64+65+66+67"  
 "+68+69+70+71+72+73+74+75+76+77+78+79+80+81+82+83+84+85+86+87+88+89+90+91+92+93+94+95+96+97+98+99+100") ==5050  
  
 assert Test\_eval("-1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20-21-22-23-24-25-26-27-28-29-30-31-32-33-34-35"  
 "-36-37-38-39-40-41-42-43-44-45-46-47-48-49-50-51-52-53-54-55-56-57-58-59-60-61-62-63-64-65-66-67"  
 "-68-69-70-71-72-73-74-75-76-77-78-79-80-81-82-83-84-85-86-87-88-89-90-91-92-93-94-95-96-97-98-99-100")==-5050  
 assert Test\_eval("1\*2\*3\*4\*5\*6\*7\*8\*9\*10\*11\*12\*13\*14\*15\*16\*17\*18\*19\*20\*21\*22\*23\*24\*25\*26\*27\*28\*29\*30\*31\*32\*33\*34\*35\*36\*"  
 "37\*38\*39\*40\*41\*42\*43\*44\*45\*46\*47\*48\*49\*50\*51\*52\*53\*54\*55\*56\*57\*58\*59\*60\*61\*62\*63\*64\*65\*66\*67\*68\*69\*"  
 "70\*71\*72\*73\*74\*75\*76\*77\*78\*79\*80\*81\*82\*83\*84\*85\*86\*87\*88\*89\*90\*91\*92\*93\*94\*95\*96\*97\*98\*99\*100")==93326215443944152681699238856266700490715968264381621468592963895217599993229915608941463976156518286253697920827223758251185210916864000000000000000000000000  
 assert Test\_eval("1/2/3/4/5/6/7/8/9/10/11/12/13/14/15/16/17/18/19/20/21/22/23/24/25/26/27/28/29/30/31/32/33/34/35/36/37/38/39/40/41/42/43/44/45/46/47/48/49/50/51/52/53/54/55/56/57/58/59/60/61/62/63/64/65/66/67/68/69/70/71/72/73/74/75/76/77/78/79/80/81/82/83/84/85/86/87/88/89/90/91/92/93/94/95/96/97/98/99/100")==1.071510288125467e-158  
 assert Test\_eval("1+2-3+4-5+6-7+8-9+10-11+12-13+14-15+16-17+18-19+20-21+22-23+24-25+26-27+28-29+30-31+32-33+34-35+36-37+38-39+40-41+42-43+44-45+46-47+48-49+50-51+52-53+54-55+56-57+58-59+60-61+62-63+64-65+66-67+68-69+70-71+72-73+74-75+76-77+78-79+80-81+82-83+84-85+86-87+88-89+90-91+92-93+94-95+96-97+98-99+100")==52  
 assert Test\_eval("1\*2/3\*4/5\*6/7\*8/9\*10/11\*12/13\*14/15\*16/17\*18/19\*20/21\*22/23\*24/25\*26/27\*28/29\*30/31\*32/33\*34/35\*36/37\*38/39\*40/41\*42/43\*44/45\*46/47\*48/49\*50/51\*52/53\*54/55\*56/57\*58/59\*60/61\*62/63\*64/65\*66/67\*68/69\*70/71\*72/73\*74/75\*76/77\*78/79\*80/81\*82/83\*84/85\*86/87\*88/89\*90/91\*92/93\*94/95\*96/97\*98/99\*100")==12.564512901854913  
 assert Test\_eval("1+2-3\*4/5")==0.6000000000000001  
if \_\_name\_\_ =="\_\_main\_\_":  
 pytest.main()

测试结果：全部通过

