**计算器**——设计开发

项目需求分析文档

2019.3.25

目录

1．引言

1．1 编写目的

1．2 文档范围

1. 项目概述

2．1 项目背景

2．2 建设目标

2．3 用户人群

2．4 运行环境

2．5 条件与限制

1. 功能需求

3．1 产品结构图

1. 引言

1．1 编写目的

•方便日常生活中所需的简单计算

1．2 文档范围

•向阅读者分析说明该程序的需求，所用到的软件，开放该程序的目的以及成品效果图。

1. 项目概述

2．1 项目背景

•名称：计算器

•代码：

# -\*- coding:utf-8 -\*-

#! python2

from tkinter import \*

class Application(Frame):

def \_\_init\_\_(self):

Frame.\_\_init\_\_(self)

self.grid()

self.mem = '' # 内存中的数据

self.opt = '' # 操作符

self.display = StringVar() # 显示的数据

self.display.set('0') # 初始值

self.need\_cls = False # 是否需要清屏

self.create\_widgets()

# 清空

def clear(self):

self.mem = ''

self.display.set('0')

# 取反

def negative(self):

self.display.set(eval('-' + self.display.get()))

# 四则运算

def option(self, opt):

if not self.need\_cls:

self.calculate()

self.opt = opt

self.need\_cls = True

self.mem = self.display.get()

# 计算结果

def calculate(self):

if self.opt:

try:

self.display.set(eval(self.mem + self.opt + self.display.get()))

except Exception:

self.display.set('错误')

self.need\_cls = True

self.opt = ''

self.mem = ''

# 百分比

def percent(self):

base = float(self.mem or 1) / 100

display = eval('{}\*{}'.format(self.display.get(), base))

int\_display = int(display)

display = int\_display if display == int\_display else display

self.display.set(display)

self.need\_cls = True

# 输入

def input(self, key):

if self.need\_cls:

self.display.set('0')

self.need\_cls = False

display = self.display.get()

if display == '0' and key != '.':

self.display.set(key)

else:

if '.' in display and key == '.':

return

self.display.set(display + key)

# 创建组件

def create\_widgets(self):

# 显示框

Entry(self, textvariable=self.display, state="readonly", width=35).grid(

row=0, column=0, columnspan=4)

# 键盘

keyboards = [

['C', '+/-', '%', '/'],

['7', '8', '9', '\*'],

['4', '5', '6', '-'],

['1', '2', '3', '+'],

['0', '.', '=']

]

for row, keys in enumerate(keyboards):

row\_num = 3 + row

for col, key in enumerate(keys):

if key == 'C':

command = self.clear

elif key == '+/-':

command = self.negative

elif key == '%':

command = self.percent

elif key in ['+', '-', '\*', '/']:

command = lambda s=key: self.option(s)

elif key == '=':

command = self.calculate

else:

command = lambda s=key: self.input(s)

bt = Button(self, text=key, command=command, width=6)

bt.grid(row=row\_num, column=col)

app = Application()

# 设置窗口标题:

app.master.title(' 计算器')

# 设置窗口尺寸/位置

app.master.geometry("326x170+200+200")

# 设置窗口不可变

app.master.resizable(width=False, height=False)

# 主消息循环:

app.mainloop()

•项目提出者：李老师

运行程序设计员：刘祖光，尚荣国

框架程序设计员：蒲慧，龙嘉雯

小组负责人：刘祖光

2．2 建设目标

•预期效果：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

2．3 用户人群

•有简单计算需求的人

2．4 运行环境

• Python3.6

1. 功能需求

3．1 项目结构图：

