Калькулятор - электронное вычислительное устройство для выполнения операций над числами или алгебраическими формулами.

Создаём окно 485 на 550. Размеры не важны, мне понравились такие. Так же указываем, что окно не будет изменяться.

Коротко о том, что делают определенные кнопки:

"C" – стирает обнуляет поле

"DEL" – убирает последнее число

"=" – равно

"/" – деление

"+" – сложение

"-" - вычитание

"1", "2", "3","4",

"5", "6","7", "8",

"9" "0" --- цифры

"+/-"- минус перед цифрой/числом

"numb^2"- возводит число в квадрат

from tkinter import \*

class Main(Frame):   
 def \_\_init\_\_(self, root):   
 super(Main, self).\_\_init\_\_(root)   
 self.build()

def build(self):

pass

def logicalc(self, operation):   
 pass

def update():   
 pass

if \_\_name\_\_ == '\_\_main\_\_':   
root = Tk()   
root["bg"] = "#000"   
root.geometry("485x550+200+200")   
root.title("Калькулятор")   
root.resizable(False, False)   
app = Main(root)   
app.pack()   
root.mainloop()

Дальше сделаем кнопки:  
btns = [ "C", "DEL", "\*", "=",

"1", "2", "3", "/", "4",

"5", "6", "+", "7", "8",

"9", "-", "+/-", "0", "%", "numb^2"

]

Он отвечает за все кнопки, отображающиеся у нас в окне.

Мы создали список, теперь проходимся циклом и отображаем эти кнопки. Для этого в том же методе пишем следующее:

x = 10

y = 140

for bt in btns:

com = lambda x=bt: self.logicalc(x)

Button(text=bt, bg="#FFF",

font=("Times New Roman", 15),

command=com).place(x=x, y=y,

width=115,

height=79)

x += 117

if x > 400:

x = 10

y += 81

Добавляем надпись с выводом результата. Я хочу что бы текст был слева, следовательно, аттрибутов выравнивания текста писать не нужно.

self.formula = "0"

self.lbl = Label(text=self.formula, font=("Times New Roman", 21, "bold"),

bg="#000", foreground="#FFF")

self.lbl.place(x=11, y=50)

Далее следует логика:  
def logicalc(self, operation):

if operation == "C":

self.formula = ""

elif operation == "DEL":

self.formula = self.formula[0:-1]

elif operation == "X^2":

self.formula = str((eval(self.formula))\*\*2)

elif operation == "=":

self.formula = str(eval(self.formula))  
 else:

if self.formula == "0":

self.formula = ""

self.formula += operation

self.update()

def update(self):

if self.formula == "":

self.formula = "0"

self.lbl.configure(text=self.formula)