**Programa CALCULADORA**

1

2 from tkinter import Tk,Text,Button,END,re

3

4 class Interfaz:

5    def \_\_init\_\_(self, ventana):

6        #Inicializar la ventana con un título

7        self.ventana=ventana

8        self.ventana.title("Calculadora Grupo 4")

9

10        #Agregar una caja de texto para que sea la pantalla de la calculadora

11        self.pantalla=Text(self.ventana, state="disabled", width=40, height=3, background="orange", foreground="white", font=("Helvetica",15))

12

13        #Ubicar la pantalla en la ventana

14        self.pantalla.grid(row=0, column=0, columnspan=4, padx=5, pady=5)

15

16        #Inicializar la operación mostrada en pantalla como string vacío

17        self.operacion=""

18

19        #Crear los botones de la calculadora

20        boton1=self.crearBoton(7)

21        boton2=self.crearBoton(8)

22        boton3=self.crearBoton(9)

23        boton4=self.crearBoton(u"\u232B",escribir=False)

24        boton5=self.crearBoton(4)

25        boton6=self.crearBoton(5)

26        boton7=self.crearBoton(6)

27        boton8=self.crearBoton(u"\u00F7")

28        boton9=self.crearBoton(1)

29       boton10=self.crearBoton(2)

30        boton11=self.crearBoton(3)

31        boton12=self.crearBoton("\*")

32        boton13=self.crearBoton(".")

33        boton14=self.crearBoton(0)

34        boton15=self.crearBoton("+")

35        boton16=self.crearBoton("-")

36        boton17=self.crearBoton("=",escribir=False,ancho=20,alto=2)

37

38        #Ubicar los botones con el gestor grid

39        botones=[boton1, boton2, boton3, boton4, boton5, boton6, boton7, boton8, boton9, boton10, boton11, boton12, boton13, boton14, boton15, boton16, boton17]

40        contador=0

41        for fila in range(1,5):

42            for columna in range(4):

43                botones[contador].grid(row=fila,column=columna)

44                contador+=1

45        #Ubicar el último botón al final

46        botones[16].grid(row=5,column=0,columnspan=4)

47

48        return

49  
50  
51    #Crea un botón mostrando el valor pasado por parámetro  
52    def crearBoton(self, valor, escribir=True, ancho=9, alto=1):  
53        return Button(self.ventana, text=valor, width=ancho, height=alto, font=("Helvetica",15), command=lambda:self.click(valor,escribir))  
54  
55  
56    #Controla el evento disparado al hacer click en un botón  
57    def click(self, texto, escribir):  
58      #Si el parámetro 'escribir' es True, entonces el parámetro texto debe mostrarse en pantalla. Si es False, no.  
59        if not escribir:  
60            #Sólo calcular si hay una operación a ser evaluada y si el usuario presionó '='  
61            if texto=="=" and self.operacion!="":  
62                #Reemplazar el valor unicode de la división por el operador división de Python '/'  
63                self.operacion=re.sub(u"\u00F7", "/", self.operacion)#re=remplaza el simbolo   
64                resultado=str(eval(self.operacion))#eval=evaluar un op matematica en un string  
65                self.operacion=""#volver a inicializar  
66                self.limpiarPantalla()  
67                self.mostrarEnPantalla(resultado)  
68            #Si se presionó el botón de borrado, limpiar la pantalla  
69            elif texto==u"\u232B":  
70                self.operacion=""  
71                self.limpiarPantalla()  
72        #Mostrar texto  
73        else:  
74            self.operacion+=str(texto)  
75            self.mostrarEnPantalla(texto)  
76        return  
77      
78  
79    #Borra el contenido de la pantalla de la calculadora   
80   def limpiarPantalla(self):  
81        self.pantalla.configure(state="normal")

82        self.pantalla.delete("1.0", END)#borra (1.0)fila y columna repectivamente[borramo s desde el inicio hasta]-> END= indice de text

83        self.pantalla.configure(state="disabled")#Vuelve a disabled

84        return

85

86

87    #Muestra en la pantalla de la calculadora el contenido de las operaciones y los resultados

88    def mostrarEnPantalla(self, valor):

89        self.pantalla.configure(state="normal")

90        self.pantalla.insert(END, valor)#inster=insertar texto en la pantalla

91        self.pantalla.configure(state="disabled")

92        return

93  
94  
95 ventana\_principal=Tk()  
96 calculadora=Interfaz(ventana\_principal)  
97 ventana\_principal.mainloop()

Para iniciar la programación de la calculadora se importa la biblioteca Tkinter y se empezara por importar “**TK**” este nos permite instanciar y crear la ventana principal en donde se mostrara la calculadora, inicializamos una clase llamada interfaz, y, dentro de ella creamos una función “*def \_\_init\_\_(self,ventana)”* donde self es una referencia a objeto mientras que ventana viene a ser el parámetro el cual representa la ventana principal la cual, después viene a instanciarse como un argumento al implementar en un objeto de tipo interfaz, creamos como parámetro la ventana principal como “*ventana\_principal=tk()*” y se creara una variable llamada calculadora donde se incluirá la interfaz y como parámetro la ventana principal obteniendo: “*calculadora=Interfaz(ventana\_principal)*”, esta entrara en un bucle infinito por medio de “*ventana\_principal.mainloop ()*” lo que permite que el programa se mantenga ejecutándose esperando eventos, es decir, las acciones que tenga el usuario en la calculadora (estas 3 últimas se ubicaran al final del código de programación, en este caso las líneas [95-97] ).

En las líneas de código (7-8) se inicializará a ventana con un título de: calculadora grupo 4, por medio de las líneas de código (10-11) agregamos una caja de texto para representarla como las pantallas de la calculadora donde se verá los números ingresados y el resultado, para ello importamos un objeto de tipo “**Text**”, por medio de la siguiente orden “*self.pantalla=Text(self.ventana, state="disabled", width=40, height=3, background="orange", foreground="white", font=("Helvetica",15))”* , en esta ocasión esta caja de texto estará ubicado dentro de la ventana la cual, estará en estado deshabilitado debido a que solo va a mostrar el resultado al usuario impidiéndole escribir en la misma, además se le da un tamaño, color de fondo, color de letra, tipo de letra;

En las líneas de código (13-14) se ubicara la pantalla dentro de la ventana, se utiliza el gestor de geometría llamado “*greed*” este gestor genera cuadriculas invisibles en la pantalla formado por filas y columnas una vez implementado obtendremos: “*self.pantalla.grid(row=0, column=0, columnspan=4, padx=5, pady=5)*” esta instrucción nos ubica a pantalla en la primera fila y columna; ”*columnspan*” indica el número de espacios que va a abarcar la pantalla, por último la función “*padx*” y “*pady*” es la posición implementando los vectores (x, y) respectivamente, esto nos permite dejar un margen en la ventana.

En las línea (17) implementamos la variable “ self.operacion=" " ” aquí vamos a guardar la operación que se va formando al momento de que el usuario utiliza a calculadora, cuando desee obtener el resultado, se evaluara dicha expresión para mostrarlo.

En las líneas (20-36) se agregará los botones que va a tener la calculadora, para lo cual se importara la clase **Button,** se creara 17 botones: 10 para los números, 4 para las operaciones fundamentales (suma, resta, multiplicación, división)**,** 3 para los símbolos (punto, borrar, igual); obteniendo “*boton1=self.crearBoton( )*” en “( )” se colocará el número que tendrá cada botón en la calculadora, pero al momento de generar el botón de borrado y división se utilizó el código Unicode para así poder representar esos dos símbolos, nos queda de la siguiente forma “*boton8=self.crearBoton(u"\u00F7")* ” donde “(u"\u00F7")” es el código del símbolo (en Unicode), hay dos botones que tienen parámetros adicionales. El primer botón es el símbolo de división el cual incluye un parámetro false esto es debido a que en Python la división se ejecuta con el símbolo “/” y no con “**÷**” que esta empleada en la calculadora el segundo botón es igual, este contiene parámetro de alto, ancho y escribir, los dos primeros parámetros generan el tamaño del botón.

Una vez creados nuestros botones en las líneas (39-46) se los ubicara en 4 filas y comunas en la pantalla utilizando “*greed*”, se colocará los botones en una lista, por medio de una estructura cíclicas for se iterará por filas y columnas, en la estructura, se introducirá la lista y se tomara cada botón mediante un contador para ubicarlos. Para el botón “=” se lo programará fuera de la iteración y en la fila 5 abarcado 4 columnas, se finalizará con un retorno para la identificar donde termina.

En las líneas 52 y 53 creamos una función para los botones, esta tendrá 4 parámetros, el primero representa el valor que va a tener cada botón, el segundo valor que indica cada botón el cual es verdadero (excepto igual y borrar), el tercer y cuarto parámetro define el tamaño de cada botón, obteniendo: “def crearBoton(self, valor, escribir=True, ancho=9, alto=1): “, en la función se retornara la instancia de **Button** pasándole parámatelos como la ventana, el valor de cada botón, su tamaño y, un comando el cual, por medio de “*lambda*” se añadira otra función que indicara que va a realizar el botón cuando se realice un “click” sobre este, mostrándolo en la pantalla obteniendo: ”return Button(self.ventana, text=valor, width=ancho, height=alto, font=("Helvetica",15), command=lambda:self.click(valor,escribir)) ”.

En las líneas (56-76) controlaremos la función de click definiremos 2 parámetros el texto, ”botón” que nos permite reconocer que botón ha sido llamado y “escribir” que indica el valor en verdadero o falso de cada botón, se implementara funciones condicionales al segundo parámetro para analizarlos, si no hay “escribir” entonces el botón presionado va a ser “igual” o “borrar” porque son falsos, si es “borrar” se eliminara todo lo que está en la pantalla (esto se lo creara en una función aparte), por otro lado si es “igual” y existe una operación, mostrara el resultado por medio de la función “*eval*”. Se importará “**re**” que es una función que nos permitirá sustituir un carácter por otro, reemplazando el símbolo “/” por “**÷**” para que Python logre interpretarlo como símbolo de división; si es verdadero el texto del botón se va a mostrar en la pantalla, para hacerlo, se agregara al atributo operación el texto del botón presionado mostrándolo en la calculadora.

En las líneas (88-92) se establecerá la función de borrar pantalla, se configurara el estado de pantalla en normal; se debe recordar que la pantalla esta deshabilitado con la finalidad de que el usuario no pueda escribir en ella, al ponerla en estado normal nos permitirá borrarla, se eliminara la pantalla desde el la fila 1, columna 0 hasta el final , para ello se importara “**END**” el cual accede al final del texto localizado dentro del objeto (text), y una vez borrada la pantalla se volverá deshabilitar la misma.

En las líneas (88-92) se creará la función mostrar pantalla la cual su parámetro será el valor que se desea ver, al igual que la función anterior se cambiará la pantalla a estado normal y en vez de borrar se implementará “*insert*” que nos permitirá insertar texto dentro de la pantalla, este será colocado al final permitiendo la concatenación de las operaciones en la calculadora y finalmente se volverá a deshabilitar la pantalla. Finamente se colocará al final del código de programación las variables mencionadas al inicio.