******

Отчёт по УП 02.01

Выполнил студент:

Зайцев Н.E.

Группа: П2-18

Преподаватель:

Тулупов А.В.

Королев 2021

Содержание отчёта

[**Раздел 1. Разработка калькулятора** 3](#_Toc88059044)

[Разработка калькулятора 3](#_Toc88059045)

[**Раздел 1. Разработка калькулятора** 4](#_Toc88059046)

[**To-Do List** 7](#_Toc88059047)

[**Заключение** 8](#_Toc88059048)

[**Приложение** 9](#_Toc88059049)

# **Раздел 1. Разработка калькулятора**

from tkinter import \*

class Main(Frame):

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

super(Main, self).\_\_init\_\_(root)

self.build()

def build(self):

self.formula = "0"

self.lbl = Label(text=self.formula, bg="#000", foreground="#FFF")

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

btns = [

"+", "-", "\*", "/",

"1", "2", "3", "DEL",

"4", "5", "6", "C",

"7", "8", "9", "=",

"0", "(", ")"

]

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

def logicalc(self, operation):

if operation == "C":

self.formula = ""

elif operation == "DEL":

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

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)

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()

# **Раздел 1. Разработка блокнота**

import os

from tkinter import Tk, Text, Menu, Scrollbar, N, S, E, W, RIGHT, Y, END

from tkinter.messagebox import showinfo

from tkinter.filedialog import asksaveasfilename, askopenfilename

class RightClicker:

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

right\_click\_menu = Menu(None, tearoff=0, takefocus=0)

for txt in ['Cut', 'Copy', 'Paste']:

right\_click\_menu.add\_command(

label=txt, command=lambda event=event, text=txt:

self.right\_click\_command(event, text))

right\_click\_menu.tk\_popup(event.x\_root, event.y\_root, entry='0')

def right\_click\_command(self, event, cmd):

event.widget.event\_generate(f'<<{cmd}>>')

class Notepad:

def \_\_init\_\_(self, \*\*kwargs):

self.file\_name = None

self.root = Tk()

self.root.title("Untitled - Notepad")

self.text\_area = Text(self.root)

menu\_bar = Menu(self.root)

file\_menu = Menu(menu\_bar, tearoff=0)

edit\_menu = Menu(menu\_bar, tearoff=0)

help\_menu = Menu(menu\_bar, tearoff=0)

self.root.config(menu=menu\_bar)

scrollbar = Scrollbar(self.text\_area)

scrollbar.pack(side=RIGHT, fill=Y)

scrollbar.config(command=self.text\_area.yview)

self.text\_area.config(yscrollcommand=scrollbar.set)

self.text\_area.bind('<Button-3>', RightClicker)

# set icon and window size (default is 300 x 300)

try:

self.root.wm\_iconbitmap("Notepad.ico")

except: #pylint: disable=W0702

pass

try:

width = kwargs['width']

except KeyError:

width = 300

try:

height = kwargs['height']

except KeyError:

height = 300

# place notepad in the center of the screen

screen\_width = self.root.winfo\_screenwidth()

screen\_height = self.root.winfo\_screenheight()

left = (screen\_width / 2) - (width / 2)

top = (screen\_height / 2) - (height / 2)

self.root.geometry('%dx%d+%d+%d' % (width, height, left, top))

self.root.grid\_rowconfigure(0, weight=1)

self.root.grid\_columnconfigure(0, weight=1)

self.text\_area.grid(sticky=N + E + S + W)

# file menu controls

file\_menu.add\_command(label="New", command=self.new\_file)

file\_menu.add\_command(label="Open", command=self.open\_file)

file\_menu.add\_command(label="Save", command=self.save\_file)

file\_menu.add\_separator()

file\_menu.add\_command(label="Exit", command=self.quit\_application)

menu\_bar.add\_cascade(label="File", menu=file\_menu)

# edit menu controls

edit\_menu.add\_command(label="Cut", command=self.cut\_text)

edit\_menu.add\_command(label="Copy", command=self.copy\_text)

edit\_menu.add\_command(label="Paste", command=self.paste\_text)

menu\_bar.add\_cascade(label="Edit", menu=edit\_menu)

# help menu controls

help\_menu.add\_command(label="About Notepad", command=self.show\_about)

menu\_bar.add\_cascade(label="Help", menu=help\_menu)

def quit\_application(self):

self.root.destroy()

def show\_about(self):

showinfo("Notepad", "Mrinal Verma")

def open\_file(self):

self.file\_name = askopenfilename(

defaultextension=".txt",

filetypes=[("All Files", "\*.\*"), ("Text Documents", "\*.txt")])

if self.file\_name == "":

self.file\_name = None

else:

self.root.title(os.path.basename(self.file\_name) + " - Notepad")

self.text\_area.delete(1.0, END)

with open(self.file\_name, 'r') as file:

self.text\_area.insert(1.0, file.read())

def new\_file(self):

self.root.title("Untitled - Notepad")

self.file\_name = None

self.text\_area.delete(1.0, END)

def save\_file(self):

if self.file\_name is None:

self.file\_name = asksaveasfilename(

initialfile='Untitled.txt', defaultextension=".txt",

filetypes=[("All Files", "\*.\*"), ("Text Documents", "\*.txt")])

if self.file\_name == "":

self.file\_name = None

else:

with open(self.file\_name, 'w') as file:

file.write(self.text\_area.get(1.0, END))

self.root.title(os.path.basename(self.file\_name) + " - Notepad")

else:

with open(self.file\_name, 'w') as file:

file.write(self.text\_area.get(1.0, END))

def cut\_text(self):

self.text\_area.event\_generate("<<Cut>>")

def copy\_text(self):

self.text\_area.event\_generate("<<Copy>>")

def paste\_text(self):

self.text\_area.event\_generate("<<Paste>>")

def run\_notepad(self):

self.root.mainloop()

def main():

notepad = Notepad(width=800, height=600)

notepad.run\_notepad()

if \_\_name\_\_ == '\_\_main\_\_':

main()

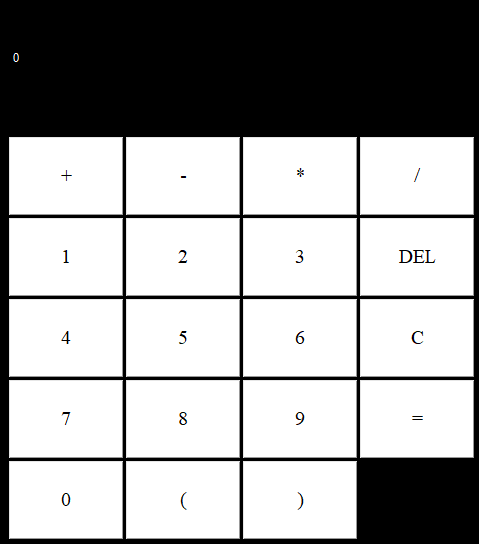
# **To-Do List**

* Провести рефакторинг кода

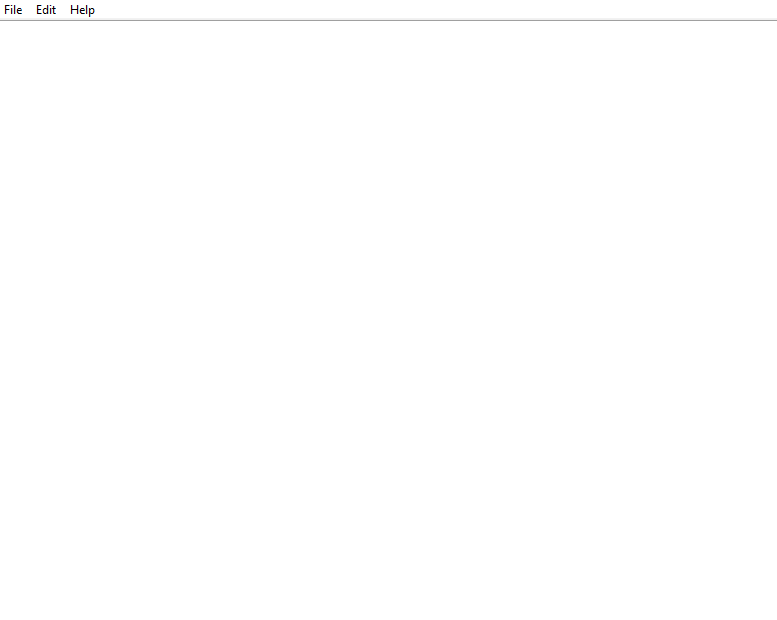
# **Заключение**

Данные программы были разработаны для использования в учебных целях изучения языка Python.

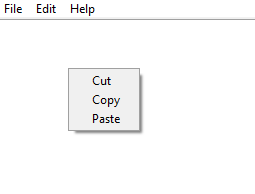
**Приложение**



**Рисунок 1**



**Рисунок 2**



**Рисунок 3**