ЗАДАЧА 1

from tkinter import \*

from tkinter.ttk import Combobox

def finish():

root.destroy() # ручное закрытие окна и всего приложения

print("Закрытие приложения")

def clicked():

lbl3.configure(text="haha")

def clicked2():

lbl4.configure(text="hehe")

root = Tk()

root.geometry("450x400")

lbl = Label(root, font=("Times New Roman Bold", 50), text="WELCOME")

lbl2 = Label(root, font=("Times New Roman", 30), text="you have to choose")

lbl3 = Label(root, font=("Times New Roman", 30))

lbl4 = Label(root, font=("Times New Roman", 30))

btn = Button(root, text="tea", font=("Times New Roman Bold", 20), bg="cyan", fg="yellow", command=clicked)

btn2 = Button(root, text="coffee", font=("Times New Roman", 20), bg="cyan", fg="red", command=clicked2)

root.title("well hello there :)")

root.protocol("WM\_DELETE\_WINDOW", finish)

lbl.grid(column=0, row=0)

lbl2.grid(column=0, row=1)

lbl3.grid(column=0, row=3)

lbl4.grid(column=2, row=3)

btn.grid(column=0, row=2)

btn2.grid(column=2, row=2)

txt = Entry(root, width=10, state='normal')

txt.focus()

txt.grid(column=0, row=4)

combo = Combobox(root)

combo['values'] = ("Black", "Green", "Ulun", "Herbal", "Milky")

combo.current(0)

combo.grid(column=0, row=5)

combo2 = Combobox(root)

combo2['values'] = ("Raf", "Americano", "Cappuccino", "Latte")

combo2.current(0)

combo2.grid(column=2, row=5)

main\_menu = Menu()

file\_menu = Menu()

file\_menu.add\_command(label='New')

file\_menu.add\_command(label='Save')

file\_menu.add\_command(label='Open')

file\_menu.add\_separator()

file\_menu.add\_command(label='Exit')

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

main\_menu.add\_cascade(label="Edit")

main\_menu.add\_cascade(label="View")

root.option\_add("\*tearOff", False)

root.config(menu=main\_menu)

root.mainloop()

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Автоматически созданное описание

ЗАДАЧА 2

import tkinter as tk

root = tk.Tk()

root.title("Basic window")

root.geometry("700x600")

root.resizable(False, False)

root.attributes("-alpha", 0.5)

frametop = tk.Frame(root,background='white',width=700, height=100)

topButton = tk.Button(frametop, text="TOP" ,cursor= "hand2" ,bg="black", fg="white", font=("Arial", 12) ,bd=0)

topButton.place(relx=0.5, rely=0.5, anchor=tk.CENTER)

frametop.pack(expand=True, fill = tk.BOTH, side=tk.TOP)

framebottom = tk.Frame(root,background='blue',width=700, height=100)

botButton = tk.Button(framebottom, text="BOTTOM" ,cursor= "hand2" ,bg="white", fg="black", font=("Arial", 12) ,bd=0)

botButton.place(relx=0.5, rely=0.5, anchor=tk.CENTER)

framebottom.pack(expand=True,fill = tk.BOTH, side=tk.BOTTOM)

frameleft = tk.Frame(root,background='yellow',width=350, height=200)

leftButton = tk.Button(frameleft, text="LEFT" ,cursor= "hand2" ,bg="white", fg="black", font=("Arial", 12) ,bd=0)

leftButton.place(relx=0.5, rely=0.5, anchor=tk.CENTER)

frameleft.pack(expand=True, side=tk.LEFT)

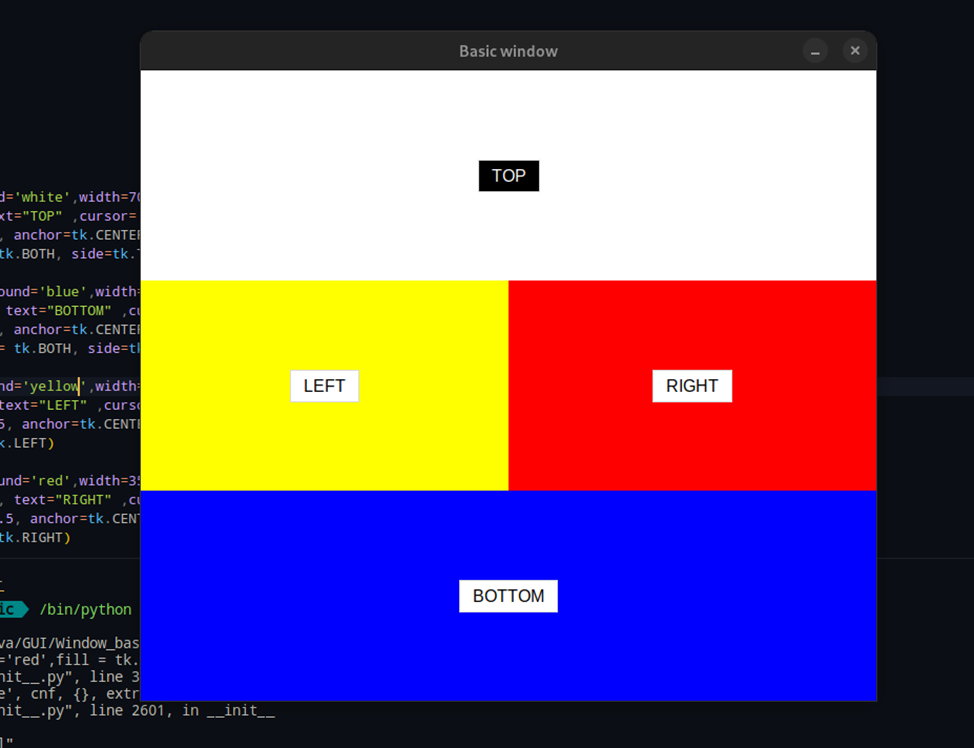
frameright = tk.Frame(root,background='red',width=350, height=200)

rightButton = tk.Button(frameright, text="RIGHT" ,cursor= "hand2" ,bg="white", fg="black", font=("Arial", 12) ,bd=0)

rightButton.place(relx=0.5, rely=0.5, anchor=tk.CENTER)

frameright.pack(expand=True, side=tk.RIGHT)

root.mainloop()



ЗАДАЧА 3

import tkinter as tk

from tkinter import ttk

def change\_entry\_color():

    color = color\_var.get()

    entry.configure(bg=color)

def change\_text\_color():

    color = text\_color\_var.get()

    entry.configure(fg=color)

def show\_text():

    text = entry.get()

    output\_label.configure(text=text)

root = tk.Tk()

root.title("Tkinter Example")

entry = tk.Entry(root, width=30)

entry.grid(row=0, column=0, padx=10, pady=10)

output\_label = tk.Label(root, text="")

output\_label.grid(row=1, column=0, padx=10, pady=10)

color\_var = tk.StringVar()

color\_var.set("white")

color\_combobox = ttk.Combobox(root, textvariable=color\_var, values=["white", "red", "green", "blue", "yellow"])

color\_combobox.grid(row=2, column=0, padx=10, pady=10)

text\_color\_var = tk.StringVar()

text\_color\_var.set("black")

text\_color\_combobox = ttk.Combobox(root, textvariable=text\_color\_var, values=["black", "red", "green", "blue", "yellow"])

text\_color\_combobox.grid(row=3, column=0, padx=10, pady=10)

change\_color\_button = tk.Button(root, text="Change Entry Color", command=change\_entry\_color)

change\_color\_button.grid(row=4, column=0, padx=10, pady=10)

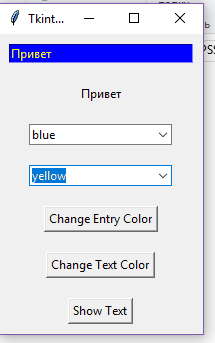
change\_text\_color\_button = tk.Button(root, text="Change Text Color", command=change\_text\_color)

change\_text\_color\_button.grid(row=5, column=0, padx=10, pady=10)

show\_text\_button = tk.Button(root, text="Show Text", command=show\_text)

show\_text\_button.grid(row=6, column=0, padx=10, pady=10)

root.mainloop()



ЗАДАЧА 4

from tkinter import \*

from datetime import datetime

temp = 0

after\_id = ''

def tick():

global temp, after\_id

after\_id = root.after(1000, tick)

f\_temp = datetime.fromtimestamp(temp).strftime("%M:%S")

label1.configure(text=str(f\_temp))

temp += 1

def start\_tick():

btnStart.pack\_forget()

btnStop.pack()

tick()

def stop\_tick():

btnStop.pack\_forget()

btnContinue.pack()

btnReset.pack()

root.after\_cancel(after\_id)

def continue\_tick():

btnContinue.pack\_forget()

btnReset.pack\_forget()

btnStop.pack()

tick()

def reset\_tick():

global temp

temp = 0

label1.configure(text='00:00')

btnContinue.pack\_forget()

btnReset.pack\_forget()

btnStart.pack()

root = Tk()

root.title('Секундомер')

root.resizable(width=False, height=False)

root.geometry('300x200')

label1 = Label(root, width=10, font=('Comic Sans MS', 30), text='00:00')

label1.pack()

btnStart = Button(root, text='Старт', font=('Comic Sans MS', 20), width=15, command=start\_tick)

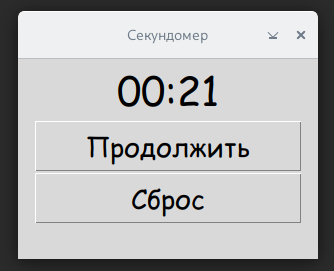
btnStop = Button(root, text='Стоп', font=('Comic Sans MS', 20), width=15, command=stop\_tick)

btnContinue = Button(root, text='Продолжить', font=('Comic Sans MS', 20), width=15, command=continue\_tick)

btnReset = Button(root, text='Сброс', font=('Comic Sans MS', 20), width=15, command=reset\_tick)

btnStart.pack()

root.mainloop()



ЗАДАЧА 5

import tkinter as tk

from tkinter import messagebox

def on\_submit():

    text = entry\_var.get()

    messagebox.showinfo("Введенный текст", text, parent=root)

def on\_color\_change():

    color = color\_var.get()

    root.config(bg=color)

root = tk.Tk()

root.title("Пример с использованием tkinter")

frame = tk.Frame(root)

frame.pack(padx=10, pady=10)

entry\_var = tk.StringVar()

entry = tk.Entry(frame, textvariable=entry\_var)

entry.pack()

button = tk.Button(frame, text="Отправить", command=on\_submit)

button.pack(pady=10)

check\_var = tk.BooleanVar()

check = tk.Checkbutton(frame, text="Чекбокс", variable=check\_var)

check.pack(pady=10)

color\_var = tk.StringVar()

color\_var.set("white")

colors = [("Белый", "white"), ("Синий", "blue"), ("Зеленый", "green")]

for text, value in colors:

    rb = tk.Radiobutton(frame, text=text, variable=color\_var, value=value, command=on\_color\_change)

    rb.pack(anchor=tk.W)

scrollbar = tk.Scrollbar(frame)

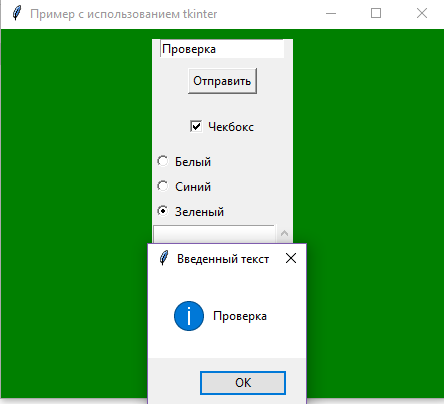
scrollbar.pack(side=tk.RIGHT, fill=tk.Y)

listbox = tk.Listbox(frame, yscrollcommand=scrollbar.set)

listbox.pack()

scrollbar.config(command=listbox.yview)

root.mainloop()



ЗАДАЧА 6

from tkinter import \*

import tkinter.font as tkFont

import tkinter as tk

class HomeHotDrinks(Tk):

    def \_\_init\_\_(self):

        super().\_\_init\_\_()

        self.title("Home Hot Drinks")

        self.geometry("720x480")

        self['background']='#664e8a'

        self.resizable(False, False)

        self.create\_widgets()

    def create\_widgets(self):

        # Creating main sign

        myFont = tkFont.Font(family="Brush Script MT", size = 36, slant = "italic", weight = "bold")

        welcomeSign = Label(text = "Home Hot Drinks", highlightthickness = 0, font = myFont, fg = "#996f50",

                       bg = "#664e8a", width = 16, height = 4)

        welcomeSign.place(x = 4, y = -10)

        # Setting application logo

        self.iconbitmap("muglogo.ico")

        # Creating logo image on the application window

        canvas = Canvas(bg = "#664e8a", width=250, height=325, highlightthickness=0)

        canvas.place(x = 5, y = 130)

        self.myImage = PhotoImage(file = "muglogo.png")

        canvas.create\_image(20, 15, anchor = NW, image = self.myImage)

        # Creating drink selector label

        menuLabelsFont1 = tkFont.Font(family="Britannic Bold", size = 14, slant = "italic", weight = "normal")

        label\_drink = tk.Label(self, text="Choose drink:", font = menuLabelsFont1, bg = "#664e8a",

                               borderwidth=2, relief="solid", width = 15, height = 3, highlightthickness=0)

        label\_drink.place(x = 350, y = 80)

        # Creating drink selector

        menuLabelsFont2 = tkFont.Font(family="Bahnschrift Condensed", size = 14, weight = "normal")

        self.drink\_options = {"Coffee": 85.99, "Tea": 45.59, "Cacao": 74.89}

        self.drink\_var = tk.StringVar(value=list(self.drink\_options.keys())[0])

        drink\_menu = tk.OptionMenu(self, self.drink\_var, \*self.drink\_options.keys())

        drink\_menu.config(font = "Georgia 14", width = 13, height = 2, bg = "#664e8a",

                           borderwidth=2, relief="solid", highlightthickness=0)

        drink\_menu.place(x = 510, y = 85)

        # Creating size selector label

        label\_size = tk.Label(self, text="Size:", font = menuLabelsFont1, bg = "#664e8a",

                               borderwidth=2, relief="solid", width = 15, height = 2, highlightthickness=0)

        label\_size.place(x = 350, y = 170)

        # Creating size selector

        self.size\_options = {"Small": 0.6, "Average": 1, "Big": 1.4}

        self.size\_var = tk.StringVar(value=list(self.size\_options.keys())[0])

        size\_menu = tk.OptionMenu(self, self.size\_var, \*self.size\_options.keys())

        size\_menu.config(font = menuLabelsFont2, width = 13, height = 1, bg = "#664e8a",

                           borderwidth=2, relief="solid", highlightthickness=0)

        size\_menu.place(x = 510, y = 170)

        # Creating extras menu label

        label\_size = tk.Label(self, text="Extras", font = menuLabelsFont1, bg = "#664e8a",

                               borderwidth=2, relief="solid", width = 15, height = 5, highlightthickness=0)

        label\_size.place(x = 350, y = 240)

        # Creating extras menu with

        # dairy cream OPTION

        self.dairy\_cream\_var = tk.BooleanVar()

        dairy\_cream\_checkbutton = tk.Checkbutton(self, text="Dairy cream", variable=self.dairy\_cream\_var)

        dairy\_cream\_checkbutton.config(font = menuLabelsFont2, width = 13, height = 0, bg = "#664e8a",

                           borderwidth=2, relief="solid", highlightthickness=0)

        dairy\_cream\_checkbutton.place(x = 510, y = 240)

        # sugar OPTION

        self.sugar\_var = tk.BooleanVar()

        sugar\_checkbutton = tk.Checkbutton(self, text="Sugar", variable=self.sugar\_var)

        sugar\_checkbutton.config(font = menuLabelsFont2, width = 13, height = 0, bg = "#664e8a",

                           borderwidth=2, relief="solid", highlightthickness=0)

        sugar\_checkbutton.place(x = 510, y = 280)

        # milk OPTION

        self.milk\_var = tk.BooleanVar()

        milk\_checkbutton = tk.Checkbutton(self, text="Milk", variable=self.milk\_var)

        milk\_checkbutton.config(font = menuLabelsFont2, width = 13, height = 0, bg = "#664e8a",

                           borderwidth=2, relief="solid", highlightthickness=0)

        milk\_checkbutton.place(x = 510, y = 320)

        # Creating quantity input field label

        label\_quantity = tk.Label(self, text="Quantity:", font = menuLabelsFont1, width = 15,

                                  height = 0, bg = "#664e8a", borderwidth=2, relief="solid", highlightthickness=0)

        label\_quantity.place(x = 350, y = 410)

        # Creating quantity input field

        self.quantity\_entry = tk.Entry(self)

        self.quantity\_entry.place(x = 510, y = 415)

        # Creating total cost label

        totalFont = tkFont.Font(family="Bodoni MT Black", size = 20, weight = "bold")

        total\_label = tk.Label(self, text="Total cost:")

        total\_label.config(font = totalFont, width = 9,

                                  height = 0, bg = "#664e8a", borderwidth=2, relief="solid", highlightthickness=0)

        total\_label.place(x = 310, y = 440)

        # Creating total cost field

        self.totalCost\_label = tk.Label(self, text=" ")

        self.totalCost\_label.config(font = totalFont, width = 13,

                                  height = 0, bg = "#664e8a", borderwidth=2, relief="solid", highlightthickness=0)

        self.totalCost\_label.place(x = 475, y = 440)

        # Creating button for making the order( calles function calculate\_total() when it's pushed)

        buttonFont = tkFont.Font(family = "Helvetica", size = 18, weight = "bold")

        button = tk.Button(self, text="Order", command=self.calculate\_total)

        button.config(font = buttonFont, width = 9,

                                  height = 0, bg = "#ab8250", borderwidth=1, relief="solid",

                                  highlightthickness=0)

        button.place(x = 450, y = 360)

    # This function calculates the total price and output information about the each order in terminal

    def calculate\_total(self):

        drink = self.drink\_var.get()

        size = self.size\_options[self.size\_var.get()]

        sizeName = "default"

        if size == 0.6:

            sizeName = "Small"

        elif size == 1:

            sizeName = "Average"

        elif size == 1.4:

            sizeName = "Big"

        quantity = int(self.quantity\_entry.get())

        price = self.drink\_options[drink]

        total = price \* size \* quantity

        sugarCount = 0

        milkCount = 0

        dairy\_creamCount = 0

        if self.sugar\_var.get():

            total += 12.99

            sugarCount = 1

        if self.milk\_var.get():

            total += 20.38

            milkCount = 1

        if self.dairy\_cream\_var.get():

            total += 33.59

            dairy\_creamCount = 1

        self.totalCost\_label.config(text="{:.2f} RUB".format(total))

        print("\nThere's one new order: ", quantity, sizeName, drink, "(-s)")

        if sugarCount or milkCount or dairy\_creamCount:

            print("\nWith extra(-s): ")

            if sugarCount:

                print(" - sugar")

            if milkCount:

                print(" - milk")

            if dairy\_creamCount:

                print(" - dairy cream")

        print("\nTotal price: {:.2f} RUB\n".format(total))

app = HomeHotDrinks()

app.mainloop()

