МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФГАОУ ВО «СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Институт информационных технологий и телекоммуникаций

Кафедра инфокоммуникаций.

Дисциплина: Технологии программирования

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №19

Основы работы с Tkinter

Выполнила:
студентка 2 курса
ИВТ-б-о-19-1
Хубиева Аида

Проверил:
Воронкин
Роман Александрович

Работа защищена с оценкой:
_____

Ставрополь, 2021

Цель работы: исследовать основы работы с Tkinter

Ход работы:

Задание 1.

Решите задачу: напишите простейший калькулятор, состоящий из двух текстовых полей,куда пользователь вводит числа, и четырех кнопок "+", "-", "*", "/". Результат вычисления должен отображаться в метке. Если арифметическое действие выполнить невозможно (например, если были введены буквы, а не числа), то в метке должно появляться слово "ошибка".

Результат выполнения задания:

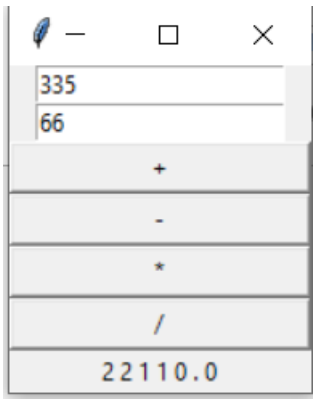


Рис.1. Результат выполнения программы

Код программы:

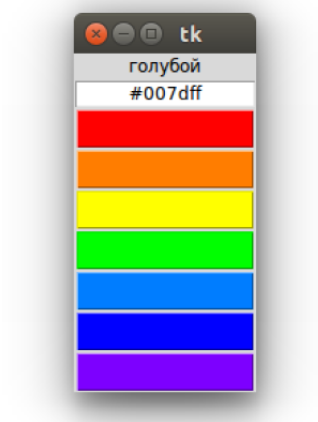
```
from tkinter import *

def func_sum(event):
    s1 = entry1.get()
    s1 = float(s1)
    s2 = entry2.get()
    s2 = float(s2)
    result1 = s1 + s2
    result1 = str(result1)
    label1['text'] = ' '.join(result1)
```

```
def func_difference(event):
    s1 = entry1.get()
    s1 = float(s1)
    s2 = entry2.get()
    s2 = float(s2)
    result2 = s1 - s2
    result2 = str(result2)
    label1['text'] = ' '.join(result2)

def func_multiplication(event):
    s1 = entry1.get()
    s1 = float(s1)
    s2 = entry2.get()
    s2 = float(s2)
    result3 = s1 * s2
    result3 = str(result3)
    label1['text'] = ' '.join(result3)

def func_splitting(event):
    s1 = entry1.get()
    s1 = float(s1)
    s2 = entry2.get()
    s2 = float(s2)
    result4 = s1 / s2
    result4 = str(result4)
    label1['text'] = ' '.join(result4)

root = Tk()

entry1 = Entry(width=20)
entry2 = Entry(width=20)
```

```
but1 = Button(width=20,text=" + ")
but2 = Button(width=20,text=" - ")
but3 = Button(width=20,text=" * ")
but4 = Button(width=20,text=" / ")
label1 = Label(width=20)
but1.bind('<Button-1>', func_sum)
but2.bind('<Button-1>', func_difference)
but3.bind('<Button-1>', func_multiplication)
but4.bind('<Button-1>', func_splitting)

entry1.pack()
entry2.pack()
but1.pack()
but2.pack()
but3.pack()
but4.pack()
label1.pack()

root.mainloop()
```

Задание 2.

Решите задачу: напишите программу, состоящую из семи кнопок, цвета которых соответствуют цветам радуги. При нажатии на ту или иную кнопку в текстовое поле должен вставляться код цвета, а в метку – название цвета.
button_off.pack(side=LEFT) root.mainloop() Коды цветов в шестнадцатеричной кодировке: #ff0000 – красный, #ff7d00 – оранжевый, #ffff00 – желтый, #00ff00 – зеленый, #007dff – голубой, #0000ff – синий, #7d00ff –
фиолетовый. Примерно должно получиться так:



Результат выполнения программы:

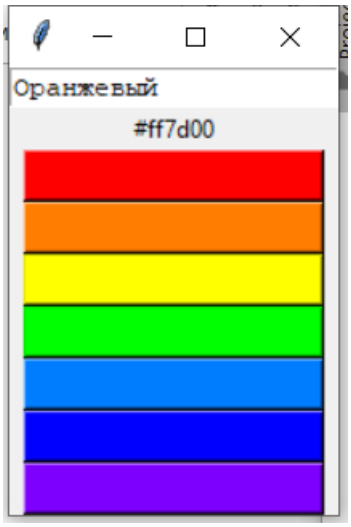


Рис.2. Результат выполнения программы

Код программы:

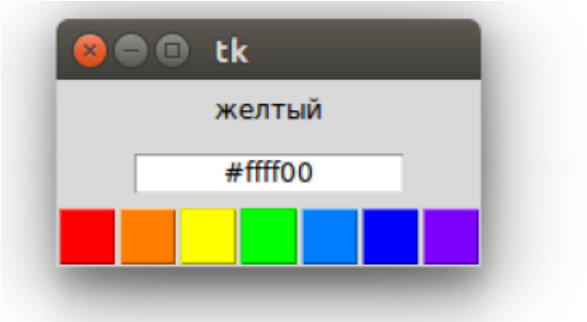
```
from tkinter import *

def color_red(event):
    text1.delete(1.0, "end")
    text1.insert(1.0,'Красный')
    label1['text'] = '#ff0000'

def color_orange(event):
    text1.delete(1.0, "end")
    text1.insert(1.0,'Оранжевый')
    label1['text'] = '#ff7d00'
```

```
def color_yellow(event):
    text1.delete(1.0, "end")
    text1.insert(1.0,'Жёлтый')
    label1['text'] = '#ffff00'

def color_green(event):
    text1.delete(1.0, "end")
    text1.insert(1.0,'Зеленый')
    label1['text'] = '#00ff00'

def color_blue(event):
    text1.delete(1.0, "end")
    text1.insert(1.0,'Голубой')
    label1['text'] = '#007dff'

def color_indigo(event):
    text1.delete(1.0, "end")
    text1.insert(1.0,'Синий')
    label1['text'] = '#0000ff'

def color_purple(event):
    text1.delete(1.0, "end")
    text1.insert(1.0,'Фиолетовый')
    label1['text'] = '#7d00ff'

root = Tk()

text1 = Text(width=20,height=1)
label1 = Label(width=20)
but1 = Button(width=20, bg='#ff0000')
but2 = Button(width=20, bg='#ff7d00')
```

```
but3 = Button(width=20, bg='#ffff00')
but4 = Button(width=20, bg='#00ff00')
but5 = Button(width=20, bg='#007dff')
but6 = Button(width=20, bg='#0000ff')
but7 = Button(width=20, bg='#7d00ff')

but1.bind('<Button-1>', color_red)
but2.bind('<Button-1>', color_orange)
but3.bind('<Button-1>', color_yellow)
but4.bind('<Button-1>', color_green)
but5.bind('<Button-1>', color_blue)
but6.bind('<Button-1>', color_indigo)
but7.bind('<Button-1>', color_purple)
text1.pack()
label1.pack()
but1.pack()
```

Задание 3.

Решите задачу: перепишите программу из пункта 8 так, чтобы интерфейс выглядел примерно следующим образом:



Результат выполнения программы:

Рис.3. Результат выполнения программы

Код программы:

```
from tkinter import *

def color_red(event):
    text1.delete(1.0, "end")
    text1.insert(1.0,'Красный')
    label1['text'] = '#ff0000'

def color_orange(event):
    text1.delete(1.0, "end")
    text1.insert(1.0,'Оранжевый')
    label1['text'] = '#ff7d00'

def color_yellow(event):
    text1.delete(1.0, "end")
    text1.insert(1.0,'Жёлтый')
    label1['text'] = '#ffff00'

def color_green(event):
    text1.delete(1.0, "end")
    text1.insert(1.0,'Зеленый')
    label1['text'] = '#00ff00'

def color_blue(event):
```

```
    text1.delete(1.0, "end")
    text1.insert(1.0,'Голубой')
    label1['text'] = '#007dff'

def color_indigo(event):
    text1.delete(1.0, "end")
    text1.insert(1.0,'Синий')
    label1['text'] = '#0000ff'

def color_purple(event):
    text1.delete(1.0, "end")
    text1.insert(1.0,'Фиолетовый')
    label1['text'] = '#7d00ff'

root = Tk()

text1 = Text(width=20,height=1)
label1 = Label(width=20)
but1 = Button(width=5, height=5, bg='#ff0000')
but2 = Button(width=5, height=5, bg='#ff7d00')
but3 = Button(width=5, height=5, bg='#ffff00')
but4 = Button(width=5, height=5, bg='#00ff00')
but5 = Button(width=5, height=5, bg='#007dff')
but6 = Button(width=5, height=5, bg='#0000ff')
but7 = Button(width=5, height=5, bg='#7d00ff')

but1.bind('<Button-1>', color_red)
but2.bind('<Button-1>', color_orange)
but3.bind('<Button-1>', color_yellow)
but4.bind('<Button-1>', color_green)
but5.bind('<Button-1>', color_blue)
but6.bind('<Button-1>', color_indigo)
```

```
but7.bind('<Button-1>', color_purple)
text1.pack()
label1.pack()
but1.pack(side='left')
but2.pack(side='left')
but3.pack(side='left')
but4.pack(side='left')
but5.pack(side='left')
but6.pack(side='left')
but7.pack(side='left')

root.mainloop()
```

Задание 4.

Решите задачу: напишите программу, состоящую из однострочного и многострочного текстовых полей и двух кнопок "Открыть" и "Сохранить". При клике на первую должен открываться на чтение файл, чье имя указано в поле класса Entry , а содержимое файла должно загружаться в поле типа Text .

При клике на вторую кнопку текст, введенный пользователем в экземпляр Text , должен сохраняться в файле под именем, которое пользователь указал в однострочном текстовом поле.

Файлы будут читаться и записываться в том же каталоге, что и файл скрипта, если указывать имена файлов без адреса. Для выполнения практической работы вам понадобится функция open языка Python и методы файловых объектов чтения и записи.

Результат выполнения задания:

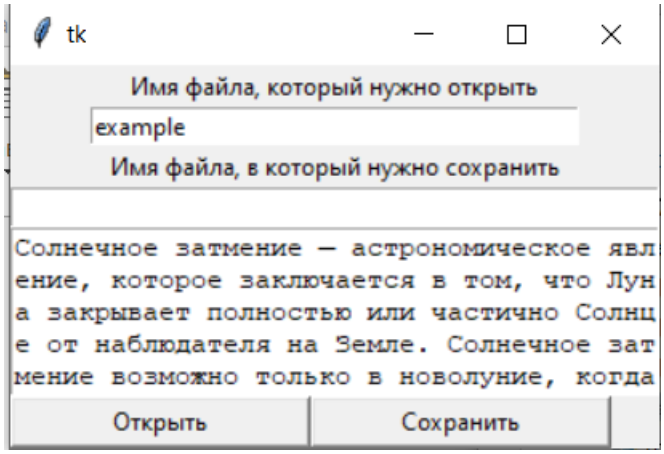


Рис. 4. Результат работы программы

Код программы:

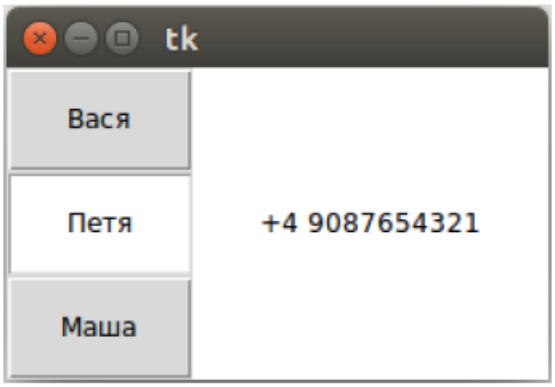
```
from tkinter import *

def open_file(event):
    s1 = entry1.get()
    s1 = str(s1)
    new_tuple = (s1, '.txt')
    complete_name = ''.join(new_tuple)
    hi = open(complete_name, 'r')
    contents_hi = hi.read()
    hi.close()
    text2.delete(1.0, "end")
    text2.insert(1.0, contents_hi)

def close_file(event):
    s2 = text2.get(1.0, "end")
    s1 = text1.get(1.0, "end-1c")
    s1 = str(s1)
    new_tuple = (s1, '.txt')
    complete_name = ''.join(new_tuple)
    bye = open(complete_name, 'w')
    bye.write(s2)
    bye.close()
```

```
root = Tk()

label1 = Label(width=40,text='Имя файла, который нужно открыть')
label2 = Label(width=40,text='Имя файла, в который нужно сохранить')
entry1 = Entry(width=40)
text1 = Text(height=1,width=40)
text2 = Text(height=5,width=40)
but1 = Button(width=20,text="Открыть")
but2 = Button(width=20,text="Сохранить")

but1.bind('<Button-1>', open_file)
but2.bind('<Button-1>', close_file)
label1.pack()
entry1.pack()
label2.pack()
text1.pack()
text2.pack()
but1.pack(side='left')
but2.pack(side='left')

root.mainloop()
```

Задание 5.

Решите задачу: виджеты Radiobatton и Checkbutton поддерживают большинство свойств оформления внешнего вида, которые есть у других элементов графического интерфейса. При этом у Radiobutton есть особое свойство indicatoron . По-умолчанию он равен единице, в этом случае радиокнопка выглядит как нормальная радиокнопка. Однако если присвоить этой опции

ноль, то виджет Radiobutton становится похожим на обычную кнопку по внешнему виду. Но не по смыслу. Напишите программу, в которой имеется несколько объединенных в группу радиокнопок, индикатор которых выключен (indicatoron=0). Если какая-нибудь кнопка включается, то в метке должна отображаться соответствующая ей информация. Обычных кнопок в окне быть не должно.



Результат выполнения задания:

Рис. 5. Результат работы программы

Код программы:

```
from tkinter import *

def lavender():
    label1['text'] = "'Lavender' - это лаванда"
    label1['bg'] = '#d4abe8'
```

```
def daisy():
    label1['text'] = "'Daisies' - ромашки"
    label1['bg'] = 'white'

def lilac():
    label1['text'] = "'Lilac' значит сирень"
    label1['bg'] = '#f7bef8'

root = Tk()
variable1 = IntVar()
variable1.set(0)
Radiobutton(text="Lavender", command=lavender,
            variable=variable1, value=0, indicatoron=0,
            width=20, bg='#d4abe8').pack()
Radiobutton(text="Daisy", command=daisy,
            variable=variable1, value=1, indicatoron=0,
            width=20, bg='white').pack()
Radiobutton(text="Lilac", command=lilac,
            variable=variable1, value=2, indicatoron=0,
            width=20, bg='#f7bef8').pack()
label1 = Label(width=20, height=10)
label1.pack(side="left")

root.mainloop()
```

Вопросы для защиты работы:

1. Какие существуют средства в стандартной библиотеке Python для построения графического интерфейса пользователя?

Используется пакет Tkinter

2. Что такое Tkinter?

Tkinter можно охарактеризовать как переводчик с языка Python на язык Tcl. Код модуля tkinter переводит программу на Python на язык Tcl, который понимает библиотека Tk.

3. Какие требуется выполнить шаги для построения графического интерфейса с помощью Tkinter?

Порядок построения GUI такой: Создать главное окно, Создать виджеты и выполнить конфигурацию их свойств, Определить события, то есть то, на что будет реагировать программа, Описать обработчики событий, то есть то, как будет реагировать программа, Расположить виджеты в главном окне, Запустить цикл обработки событий.

4. Что такое цикл обработки событий?

Цикл обработки событий приводит к отображению главного окна со всеми упакованными на нем виджетами.

5. Каково назначение экземпляра класса Tk при построении графического интерфейса с помощью Tkinter?

Каждый экземпляра класса Tk создает новое окно.

6. Для чего предназначены виджеты Button, Label, Entry и Text?

Button - создание кнопки, Label - создание текстовой метки, Entry - текстовое поле для ввода и Text - текстовое поле.

7. Каково назначение метода pack() при построении графического интерфейса пользователя?

Расположение всех виджетов окна в самом простом порядке, друг под другом.

8. Как осуществляется управление размещением виджетов с помощью метода pack()?

Метод pack() помещается в программе перед инициатором главного цикла обработки событий и располагает виджеты окна в виде столбца.

9. Как осуществляется управление полосами прокрутки в виджете Text?

Для создания полосы прокрутки есть отдельный тип виджета Scrollbar, который в программе располагается после Text.

10. Для чего нужны тэги при работе с виджетом Text?

Для изменения шрифта, цвета, ширины и высоты виджета.

11. Как осуществляется вставка виджетов в текстовое поле?

В Text можно вставлять другие виджеты помощью метода window_create

12. Для чего предназначены виджеты Radiobutton и Checkbutton?

Для создания радиокнопки и флажков.

13. Что такое переменные Tkinter и для чего они нужны?

Переменные Tkinter используются для виджета Checkbutton. Они нужны, чтобы снимать сведения о состоянии флажков.

14. Как осуществляется связь переменных Tkinter с виджетами Radiobutton и Checkbutton?

У каждого флажка должна быть своя переменная Tkinter, у которой должны быть методы onvalue и offvalue, значения которых определяются как 0 и 1.