

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное автономное
образовательное учреждение высшего образования
«Северо-Кавказский федеральный университет»**

Кафедра инфокоммуникаций

**Отчет по лабораторной работе № 4.7
«Основы работы с Tkinter»**

по дисциплине «Объектно-ориентированное программирование»

Выполнил студент группы ИВТ-б-о-20-1

Злыгостев И.С. « » _____ 20__ г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил Воронкин Р.А. _____
(подпись)

Ставрополь 2022

Цель работы: приобретение навыков построения графического интерфейса пользователя GUI с помощью пакета Tkinter языка программирования Python версии 3.x.

Ход работы:

1. Клонировал общедоступный репозиторий GitHub на свой локальный сервер.
2. Изучил теоретический материал и приступил к выполнению заданий.

```
15 def add(event):
16     try:
17         num1 = float(ent1.get())
18         num2 = float(ent2.get())
19         l1['text'] = num1+num2
20     except ValueError:
21         l1['text'] = 'Ошибка'
22
23
24 def sub(event):
25     try:
26         num1 = float(ent1.get())
27         num2 = float(ent2.get())
28         l1['text'] = num1-num2
29     except ValueError:
30         l1['text'] = 'Ошибка'
31
32
33 def mul(event):
34     try:
35         num1 = float(ent1.get())
36         num2 = float(ent2.get())
37         l1['text'] = num1*num2
38     except ValueError:
```

Рисунок 1 – Код первого задания

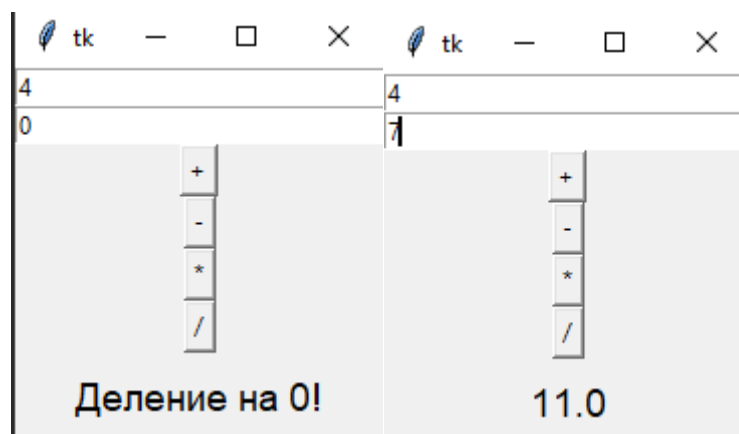


Рисунок 2 – Результат работы программы

3. Приступил к выполнению второго задания.

```

14 def red(event):
15     ent1.delete(0, END)
16     l1['text'] = "Красный"
17     ent1.insert(0, "#ff0000")
18
19
20 def orange(event):
21     ent1.delete(0, END)
22     l1['text'] = "Оранжевый"
23     ent1.insert(0, "#ff7d00")
24
25
26 def yellow(event):
27     ent1.delete(0, END)
28     l1['text'] = "Жёлтый"
29     ent1.insert(0, "#ffff00")
30

```

Рисунок 3 – Фрагмент кода второго задания

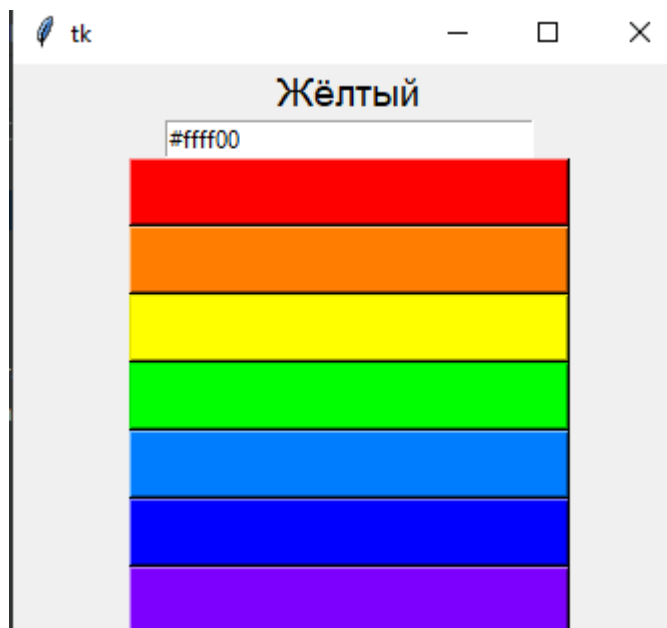


Рисунок 4 – Результат выполнения программы

4. Выполнил третье задание.

```

55 l1 = Label(font="Arial 14", width=30)
56 ent1 = Entry(width=30)
57 but1 = Button(bg='#ff0000', width=5, pady=5)
58 but2 = Button(bg='#ff7d00', width=5, pady=5)
59 but3 = Button(bg='#ffff00', width=5, pady=5)
60 but4 = Button(bg='#00ff00', width=5, pady=5)
61 but5 = Button(bg='#007dff', width=5, pady=5)
62 but6 = Button(bg='#0000ff', width=5, pady=5)
63 but7 = Button(bg='#7d00ff', width=5, pady=5)
64 but1.bind('<Button-1>', red)
65 but2.bind('<Button-1>', orange)
66 but3.bind('<Button-1>', yellow)
67 but4.bind('<Button-1>', green)
68 but5.bind('<Button-1>', gol)
69 but6.bind('<Button-1>', blue)
70 but7.bind('<Button-1>', fiol)
71 l1.pack()
72 ent1.pack()
73 but1.pack(side=LEFT, padx=3, pady=5)
74 but2.pack(side=LEFT, padx=3, pady=5)
75 but3.pack(side=LEFT, padx=3, pady=5)
76 but4.pack(side=LEFT, padx=3, pady=5)
77 but5.pack(side=LEFT, padx=3, pady=5)
78 but6.pack(side=LEFT, padx=3, pady=5)
79 but7.pack(side=LEFT, padx=3, pady=5)
80 root.mainloop()

```

Рисунок 5 – Фрагмент кода третьего задания

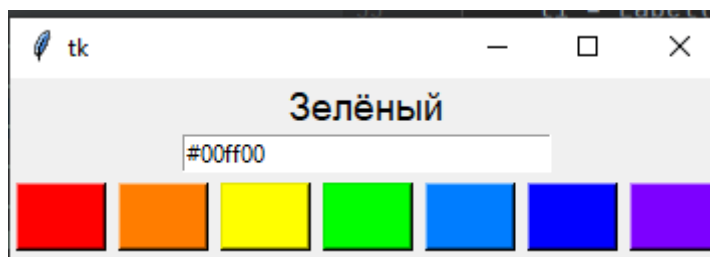


Рисунок 6 – Результат выполнения задания

5. Выполнил четвертое задание.

```

15 def save(event):
16     name = ent.get()
17     data = text.get(1.0, END)
18     with open(name, 'w', encoding="utf-8") as f:
19         f.write(data)
20
21
22 def opening(event):
23     try:
24         text.delete(1.0, END)
25         name = ent.get()
26         with open(name, 'r', encoding="utf-8") as f:
27             data = f.read()
28         text.insert(1.0, data)
29     except FileNotFoundError:
30         text.insert(1.0, 'Укажите путь к файлу')
31

```

Рисунок 7 – Фрагмент кода четвертого задания

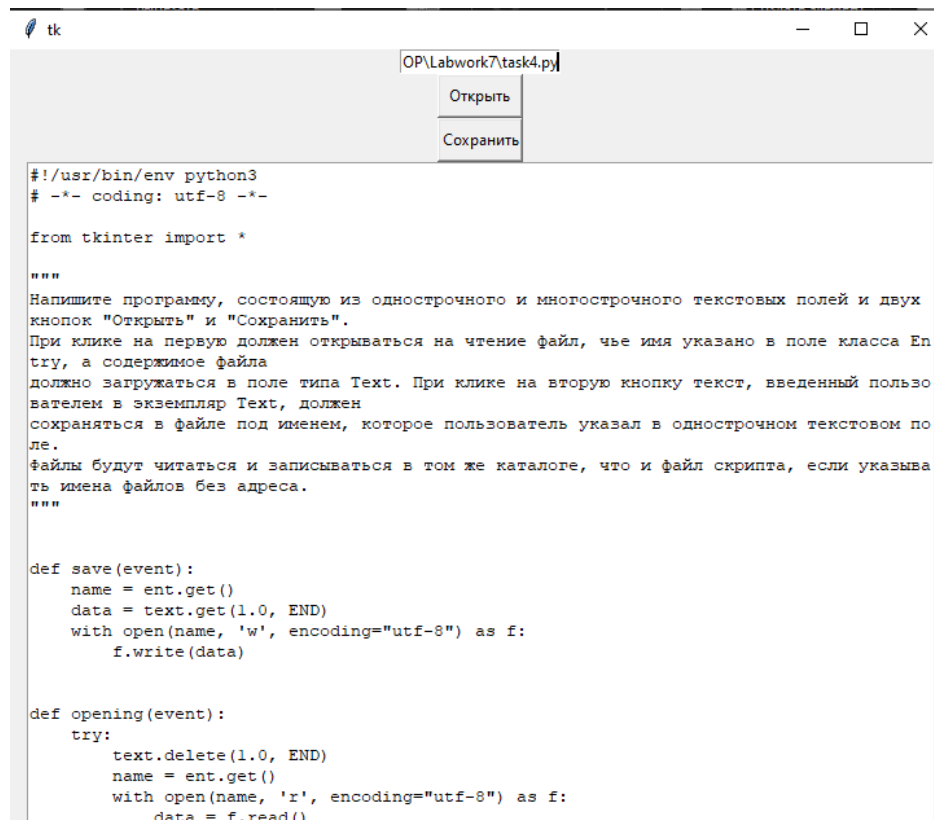


Рисунок 8 – Результат выполнения программы

6. Выполнил пятое задание.

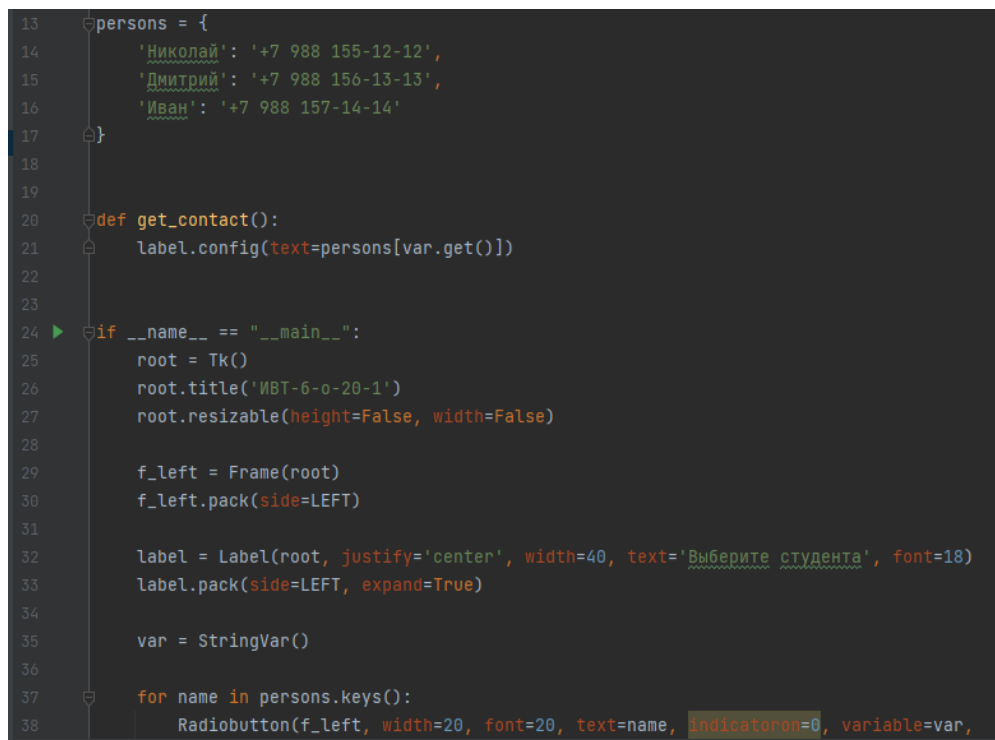


Рисунок 9 – Фрагмент кода пятого задания

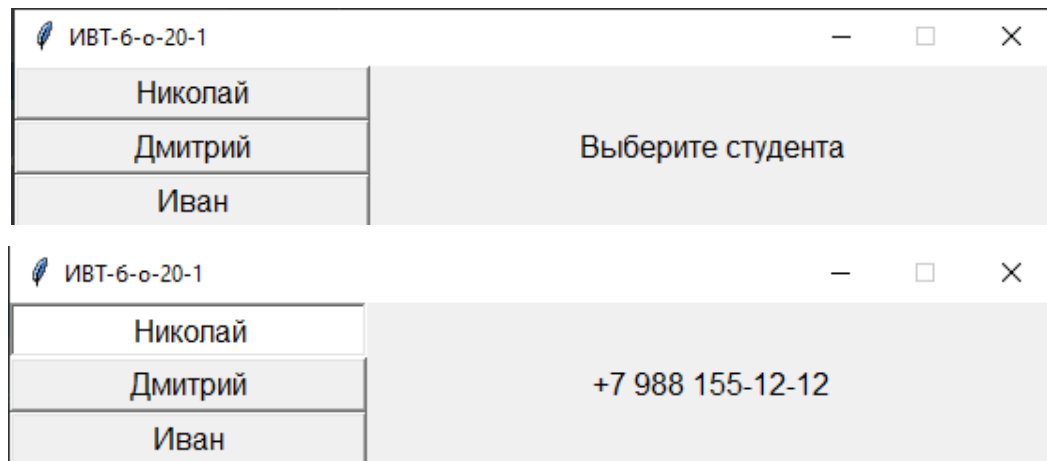


Рисунок 10 – Работа с графическим интерфейсом в пятом задании

Контрольные вопросы:

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

Существует множество библиотек GUI, среди которых Tk не самый популярный инструмент, хотя с его помощью написано немало проектов. Он был выбран для Python по-умолчанию. Установочный файл интерпретатора Питона обычно уже включает пакет tkinter в составе стандартной библиотеки.

2. Что такое Tkinter?

Tkinter – это пакет для Python, предназначенный для работы с библиотекой Tk. Библиотека Tk содержит компоненты графического интерфейса пользователя (graphical user interface – GUI), написанные на языке программирования Tcl.

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

- Создать главное окно.
 - Создать виджеты и выполнить конфигурацию их свойств (опций).
 - 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?

Tkinter является событийно-ориентированной библиотекой. В приложениях такого типа имеется главный цикл обработки событий. В Tkinter такой цикл запускается методом `mainloop`. Для явного выхода из интерпретатора и завершения цикла обработки событий используется метод `quit`.

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

Метод `mainloop` экземпляра Tk запускает главный цикл обработки событий, что в том числе приводит к отображению главного окна со всеми "упакованными" на нем виджетами

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

Button – кнопка. Самыми важными свойствами виджета класса Button являются `text`, с помощью которого устанавливается надпись на кнопке, и `command` для установки действия, то есть того, что будет происходить при нажатии на кнопку.

Label – метка Виджет Label просто отображает текст в окне и служит в основном для информационных целей (вывод сообщений, подпись других элементов интерфейса).

Entry – однострочное текстовое поле Текстовые поля предназначены для ввода информации пользователем.

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

Метод «Pack()» - упаковщик. Ранее мы его уже использовали для отображения наших виджетов в окне программы, но использовали без дополнительных параметров. И по умолчанию наши виджеты располагались друг под другом, в зависимости в какой последовательности был применен метод «`pack()`» к объектам.

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

Если в упаковщики не передавать аргументы, то виджеты будут располагаться вертикально, друг над другом. Тот объект, который первым вызовет `pack`, будет вверху. Который вторым – под первым, и так далее.

У метода `pack` есть параметр `side` (сторона), который принимает одно из четырех значений-констант `tkinter` – `TOP`, `BOTTOM`, `LEFT`, `RIGHT` (верх, низ, лево, право). По умолчанию, когда в `pack` не указывается `side`, его значение равняется `TOP`. Из-за этого виджеты располагаются вертикально.

9. Как осуществляется управление полосами прокрутки в виджете `Text`?

Если в текстовое поле вводится больше линий текста, чем его высота, то оно само будет прокручиваться вниз. При просмотре прокручивать вверх-вниз можно с помощью колеса мыши и стрелками на клавиатуре. Однако бывает удобнее пользоваться скроллером – полосой прокрутки. В `tkinter` скроллеры производятся от класса `Scrollbar`. Объект-скроллер связывают с виджетом, которому он требуется. Это не обязательно многострочное текстовое поле.

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

Особенностью текстового поля библиотеки `Tk` является возможность форматировать текст в нем, то есть придавать его разным частям разное оформление. Делается это с помощью методов `tag_add` и `tag_config`. Первый добавляет тег, при этом надо указать его произвольное имя и отрезок текста, к которому он будет применяться. Метод `tag_config` настраивает тегу стили оформления.

11. Как осуществляется вставка виджетов в текстовое поле?

В `Text` можно вставлять другие виджеты помощью метода `window_create`. Потребность в этом не велика, однако может быть интересна с объектами типа `Canvas`.

12. Для чего предназначены виджеты `Radiobutton` и `Checkbutton`?

`Checkbutton` – это виджет, который позволяет отметить „галочкой“ определенный пункт в окне. При использовании нескольких пунктов нужно каждому присвоить свою переменную; `Radiobutton` выполняет функцию,

схожую с функцией виджета Checkbutton. Разница в том, что в виджете Radiobutton пользователь может выбрать лишь один из пунктов.

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

В Tkinter нельзя использовать любую переменную для хранения состояний виджетов. Для этих целей предусмотрены специальные классы-переменные пакета tkinter – BooleanVar, IntVar, DoubleVar, StringVar. Первый класс позволяет принимать своим экземплярам только булевы значения (0 или 1 и True или False), второй – целые, третий – дробные, четвертый – строковые.

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

При запуске программы включенной окажется первая радиокнопка, так как значение ее опции value совпадает с текущим значением переменной r_var. Если кликнуть по второй радиокнопке, то она включится, а первая выключится. При этом значение r_var станет равным 1. В функции change в зависимости от считанного значения переменной var ход выполнения программы идет по одной из трех веток.

Вывод: в ходе выполнения лабораторной работы были приобретены навыки построения графического интерфейса пользователя GUI с помощью пакета Tkinter языка программирования Python версии 3.x.