

**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-21-1

Лысенко И.А. « » _____ 20__ г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил Воронкин Р.А. _____

(подпись)

Ставрополь 2024

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

Ход работы:

1. Создал репозиторий на GitHub:

https://github.com/IsSveshuD/OOP_Lab_4.7.git .

2. Выполнил задание 1.

```
import tkinter as tk

"""
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".
"""

def calculate(operation):
    try:
        num1 = float(text1.get())
        num2 = float(text2.get())

        if operation == "+":
            result = num1 + num2
        elif operation == "-":
            result = num1 - num2
        elif operation == "*":
            result = num1 * num2
        elif operation == "/":
            if num2 == 0:
                result = "Ошибка"
            else:
                result = num1 / num2
        else:
            result = "Ошибка"

        result1.config(text=f"{result}")
    except ValueError:
        result1.config(text="Ошибка")

root = tk.Tk()
root.title("Калькулятор")

text1 = tk.Entry(root)
text1.grid(row=0)
text2 = tk.Entry(root)
text2.grid(row=1)
result1 = tk.Label(root, text="Результат:")
result1.grid(row=7, pady=10)
add_btn = tk.Button(root, text="+", width=10, command=lambda: calculate("+"))
add_btn.grid(row=3, pady=5)
sub_btn = tk.Button(root, text="-", width=10, command=lambda: calculate("-"))
sub_btn.grid(row=4, pady=5)
mul_btn = tk.Button(root, text="*", width=10, command=lambda: calculate("*"))
mul_btn.grid(row=5, pady=5)
div_btn = tk.Button(root, text="/", width=10, command=lambda: calculate("/"))
div_btn.grid(row=6, pady=5)

root.mainloop()
```

Рисунок 1 – Задание 1

3. Получил следующий результат работы задания 1.



Рисунок 2 – Результат работы задания 1

4. Выполнил задание 2.

```
import tkinter as tk

"""
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. Коды цветов в шестнадцатеричной
кодировке: #ff0000 - красный, #ff7d00 - оранжевый, #ffff00 -
желтый, #00ff00 - зеленый, #007dff - голубой, #0000ff - синий,
#7d00ff - фиолетовый.
"""

color_codes = {
    "#ff0000": "Красный",
    "#ff7d00": "Оранжевый",
    "#ffff00": "Желтый",
    "#00ff00": "Зелёный",
    "#007dff": "Голубой",
    "#0000ff": "Синий",
    "#7d00ff": "Фиолетовый",
}

def button_click(color_code):
    col_label.config(text=color_codes[color_code])
    text.delete(1.0, tk.END)
    text.insert(tk.END, color_code, "centered")
```

```

root = tk.Tk()
root.title("Цвета радуги")
col_label = tk.Label(root)
col_label.pack(pady=5)
text = tk.Text(root, height=1, width=15)
text.pack(pady=5)
text.tag_configure("centered", justify="center")

colors = ["#ff0000", "#ff7d00", "#ffff00", "#00ff00",
          "#007dff", "#0000ff", "#7d00ff"]
for color in colors:
    btn = tk.Button(root, bg=color, width=15, height=2,
                    command=lambda c=color: button_click(c))
    btn.pack(pady=2)

root.mainloop()

```

Рисунок 3 – Задание 2

5. Получил следующий результат работы задания 2.



Рисунок 4 – Результат работы задания 2

6. Выполнил задание 3.

```

import tkinter as tk

color_codes = {
    "#ff0000": "Красный",
    "#ff7d00": "Оранжевый",
    "#ffff00": "Желтый",
    "#00ff00": "Зелёный",
    "#007dff": "Голубой",
    "#0000ff": "Синий",
    "#7d00ff": "Фиолетовый",
}

def button_click(color_code):
    col_label.config(text=color_codes[color_code])
    text.delete(1.0, tk.END)
    text.insert(tk.END, color_code, "centered")

root = tk.Tk()
root.title("Цвета радуги")
col_label = tk.Label(root)
col_label.pack(pady=5)
text = tk.Text(root, height=1, width=30)
text.pack(pady=5)
text.tag_configure("centered", justify="center")

btn_frame = tk.Frame(root)
btn_frame.pack(pady=10)
colors = ["#ff0000", "#ff7d00", "#ffff00", "#00ff00",
          "#007dff", "#0000ff", "#7d00ff"]
for color in colors:
    btn = tk.Button(btn_frame, bg=color, width=5, height=2,
                    command=lambda c=color: button_click(c))
    btn.pack(side=tk.LEFT, padx=2)

root.mainloop()

```

Рисунок 5 – Задание 3

7. Получил следующий результат работы задания 3.

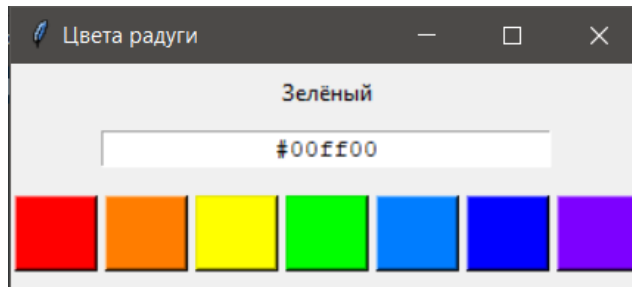


Рисунок 6 – Результат работы задания 3

8. Выполнил задание 4.

```
import tkinter as tk
from tkinter import messagebox
import os

"""
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.
"""

class FileEditor:
    def __init__(self, root):
        self.root = root
        self.root.title("Текстовый редактор")
        self.filename = tk.Entry(root, width=40)
        self.filename.grid(row=0, column=0, padx=5, pady=5)

        open_btn = tk.Button(root, text="Открыть", command=self.open_file)
        open_btn.grid(row=0, column=1, padx=5, pady=5)

        save_btn = tk.Button(root, text="Сохранить", command=self.save_file)
        save_btn.grid(row=0, column=2, padx=5, pady=5)

        self.text_editor = tk.Text(root, height=10, width=50)
        self.text_editor.grid(row=1, column=0, columnspan=3, padx=5, pady=5)
```

```

def open_file(self):
    filename = self.filename.get()
    try:
        with open(os.path.join(os.path.dirname(__file__),
                                filename), 'r') as file:
            content = file.read()
            self.text_editor.delete(1.0, tk.END)
            self.text_editor.insert(tk.END, content)
    except FileNotFoundError:
        messagebox.showerror("Ошибка", "Файл не найден")

def save_file(self):
    filename = self.filename.get()
    filepath = os.path.join(os.path.dirname(__file__), filename)
    content = self.text_editor.get(1.0, tk.END)
    with open(filepath, 'w') as file:
        file.write(content)
    messagebox.showinfo("Результат", "Файл сохранен")

if __name__ == "__main__":
    root = tk.Tk()
    app = FileEditor(root)
    root.mainloop()

```

Рисунок 7 – Задание 4

9. Получил следующий результат работы задания 4.

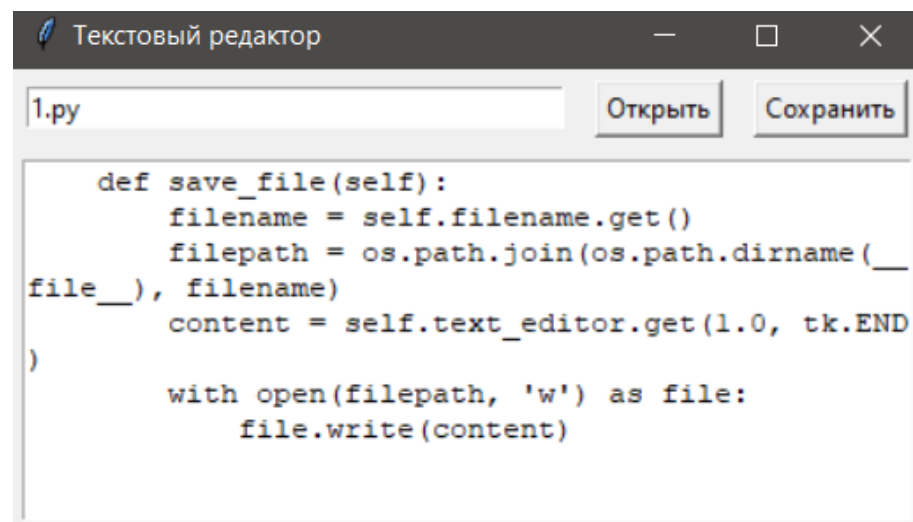


Рисунок 8 – Результат работы задания 4

10. Выполнил задание 5.

```

import tkinter as tk

"""
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.
"""

def update_label():
    selected_option = var.get()
    label.config(text=f"{selected_option}")

root = tk.Tk()
root.title("Кнопки")

var = tk.StringVar()
radio_btn_1 = tk.Radiobutton(root, text="Вася", width=10,
                              height=3, variable=var, value="Рыбы",
                              indicatoron=False, command=update_label)
radio_btn_2 = tk.Radiobutton(root, text="Дина", width=10,
                              height=3, variable=var, value="Козерог",
                              indicatoron=False, command=update_label)
radio_btn_3 = tk.Radiobutton(root, text="Лиза", width=10,
                              height=3, variable=var, value="Весы",
                              indicatoron=False, command=update_label)

label = tk.Label(root, width=20, anchor="center")
label.grid(row=0, column=1, rowspan=3, sticky="w")
radio_btn_1.grid(row=0, column=0, sticky="w")
radio_btn_2.grid(row=1, column=0, sticky="w")
radio_btn_3.grid(row=2, column=0, sticky="w")

root.mainloop()

```

Рисунок 9 – Задание 5

11. Получил следующий результат работы задания 5.

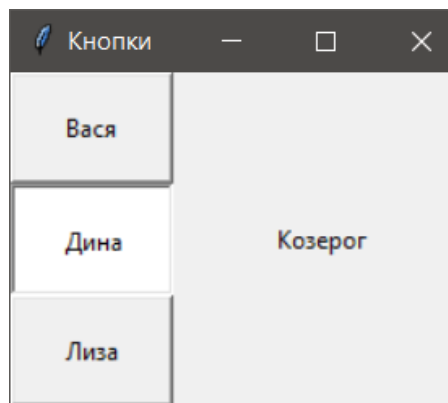


Рисунок 10 – Результат работы задания 5

Ответы на вопросы:

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

Основным средством для создания графического интерфейса в стандартной библиотеке Python является модуль `tkinter`.

2. Что такое Tkinter?

Tkinter — это стандартная библиотека Python для создания графического интерфейса пользователя. Она предоставляет набор инструментов и виджетов для построения окон, кнопок, текстовых полей, меток и других элементов управления.

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

Шаги для построения графического интерфейса с помощью Tkinter:

- 1) Создание главного окна (`Tk`).
- 2) Создание и настройка виджетов (кнопок, меток и т.д.).
- 3) Упаковка или размещение виджетов в окне с использованием метода `pack()`, `grid()` или `place()`.
- 4) Запуск цикла обработки событий (`mainloop()`).
4. Что такое цикл обработки событий?

Цикл обработки событий — это бесконечный цикл, который ожидает и обрабатывает события, такие как нажатия кнопок, перемещения мыши и другие взаимодействия с пользователем. В Tkinter это обеспечивает метод `mainloop()`.

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

Экземпляр класса `Tk` представляет главное окно приложения. Его цель — создать основное окружение для построения графического интерфейса с использованием Tkinter.

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

Button: Кнопка, предназначенная для выполнения действия п нажатии.

Label: Метка для отображения текста или изображения.

Entry: Однострочное текстовое поле для ввода данных.

Text: Многострочное текстовое поле для ввода и отображения текста.

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

Метод pack() используется для размещения виджетов в родительском контейнере. Он автоматически управляет размерами виджетов и их расположением в окне.

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

side: Задает сторону (верх, низ, лево, право), на которую будет упакован виджет.

fill: Определяет, как виджет заполняет доступное пространство ("x", "y", "both", "none").

expand: Указывает, следует ли расширять виджет для заполнения доступного пространства.

9. Как осуществляется управление полосами прокрутки в виджете Text?

Для управления полосами прокрутки в виджете Text используются виджеты Scrollbar и их методы привязки (yview и xview).

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

Тэги в виджете Text используются для применения форматирования, стилей и связывания событий к определенным частям текста.

11. Как осуществляется вставка виджетов в текстовое поле?

Для вставки виджетов, таких как кнопки и изображения, в текстовое поле используются методы window_create() и insert().

12. Для чего предназначены виджеты Radiobutton и Checkbutton?

Radiobutton: Позволяет пользователю выбирать один из нескольких взаимоисключающих вариантов.

Checkbutton: Предоставляет пользователю возможность включать или выключать определенные опции, независимо друг от друга.

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

Переменные Tkinter, такие как StringVar, IntVar и DoubleVar, предоставляют связь между значениями переменных и виджетами, что позволяет автоматически обновлять виджеты при изменении переменных.

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

Для связи с Radiobutton используется параметр variable с объектом переменной (StringVar, IntVar и т.д.).

Для связи с Checkbutton используется параметр variable с объектом IntVar, который принимает 0 или 1 в зависимости от состояния флажка.

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