

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ
ФЕДЕРАЦИИ**
**Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРОКАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

Кафедра инфокоммуникаций

Институт цифрового развития

ОТЧЁТ

по лабораторной работе

Дисциплина: «Объектно – ориентированное программирование»

Выполнил студент группы

ИВТ-б-о-21-1

Богдашов А.В. « » _____ 20__ г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил доцент

Кафедры инфокоммуникаций, старший
преподаватель

Воронкин Р.А.

(подпись)

Основы работы с tkinter

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

Ход работа:

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".

Код программы:

```
import tkinter as tk

class Calculator:
    def __init__(self, roote):
        self.root = roote
        self.root.title("Простой калькулятор")

        self.num1_entry = tk.Entry(roote, width=10)
        self.num1_entry.grid(row=0, column=0, padx=5, pady=5)

        self.operator_label = tk.Label(roote, text="", width=2)
        self.operator_label.grid(row=0, column=1, padx=5, pady=5)

        self.num2_entry = tk.Entry(roote, width=10)
        self.num2_entry.grid(row=0, column=2, padx=5, pady=5)

        self.result_label = tk.Label(roote, text="Результат: ", width=20)
        self.result_label.grid(row=1, column=0, columnspan=3, padx=5, pady=5)

        self.operations = {
            "сложение": "+",
            "вычитание": "-",
            "умножение": "*",
            "деление": "/"
        }

        self.add_button = tk.Button(roote, text="Сложение", command=lambda:
self.calculate("сложение"))
        self.add_button.grid(row=2, column=0, padx=5, pady=5)

        self.subtract_button = tk.Button(roote, text="Вычитание", command=lambda:
self.calculate("вычитание"))
        self.subtract_button.grid(row=2, column=1, padx=5, pady=5)

        self.multiply_button = tk.Button(roote, text="Умножение", command=lambda:
self.calculate("умножение"))
        self.multiply_button.grid(row=2, column=2, padx=5, pady=5)

        self.divide_button = tk.Button(roote, text="Деление", command=lambda:
self.calculate("деление"))
        self.divide_button.grid(row=3, column=0, padx=5, pady=5)

    def calculate(self, operation):
        try:
            num1 = float(self.num1_entry.get())
            num2 = float(self.num2_entry.get())
```

```

        if operation in self.operations:
            operator = self.operations[operation]
            result = eval(f"{num1} {operator} {num2}")

            self.operator_label.config(text=operation)
            self.result_label.config(text=f"Результат: {result}")
        else:
            self.result_label.config(text="Ошибка: Неподдерживаемая операция")

    except ValueError:
        self.result_label.config(text="Ошибка: введите число")

if __name__ == "__main__":
    root = tk.Tk()
    calculator = Calculator(root)
    root.mainloop()

```

Результат работы программы:

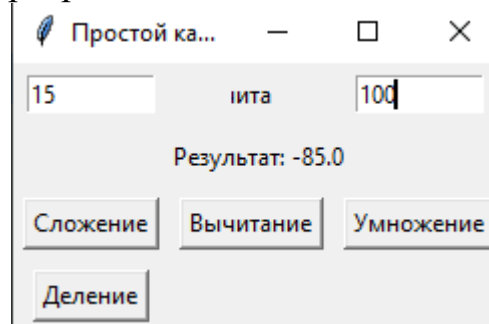


Рисунок 1. Результат работы программы к первому заданию

Задание 2.

Код программы:

```

import tkinter as tk

def update_color(color_codes, color_names):
    color_code_label.config(text=f"Код цвета: {color_codes}")
    color_name_label.config(text=f"Название цвета: {color_names}")

def button_clicked(color_codes, color_names):
    update_color(color_codes, color_names)

root = tk.Tk()
root.title("Rainbow Colors")

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

color_code_label = tk.Label(root, text="Код цвета: ")
color_code_label.pack()

color_name_label = tk.Label(root, text="Название цвета: ")
color_name_label.pack()

```

```

for color_name, color_code in colors.items():
    button = tk.Button(root, text=color_name, bg=color_code, width=10,
                       command=lambda c=color_code, n=color_name: button_clicked(c,
n))
    button.pack(pady=5)

root.mainloop()

```

Результат работы программы:



Рисунок 2. Результат работы программы ко второму заданию

Задание 3.

Код программы:

```

import tkinter as tk

def update_color(color_codes, color_names):
    color_code_label.config(text=f"Код цвета: {color_codes}")
    color_name_label.config(text=f"Название цвета: {color_names}")

def button_clicked(color_codes, color_names):
    update_color(color_codes, color_names)

root = tk.Tk()
root.title("Rainbow Colors")

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

color_code_label = tk.Label(root, text="Код цвета: ")
color_code_label.grid(row=0, column=0, columnspan=len(colors), pady=5)

color_name_label = tk.Label(root, text="Название цвета: ")

```

```

color_name_label.grid(row=1, column=0, columnspan=len(colors), pady=5)

column_counter = 0
for color_name, color_code in colors.items():
    button = tk.Button(root, text=color_name, bg=color_code, width=10,
                       command=lambda c=color_code, n=color_name: button_clicked(c,
n))
    button.grid(row=2, column=column_counter, padx=5)
    column_counter += 1

root.mainloop()

```

Результат работы программы:

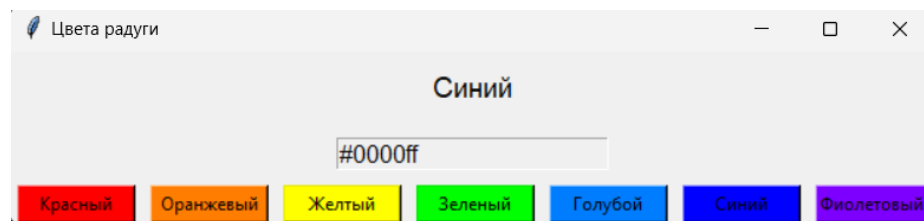


Рисунок 3. Результат работы программы к третьей задачи

Задание 4.

Код программы:

```

import tkinter as tk
import os

class FileEditorApp:
    def __init__(self, roote):
        self.root = roote
        self.root.title("Текстовый редактор")

        self.entry_filename = tk.Entry(roote, width=30)
        self.entry_filename.grid(row=0, column=0, padx=5, pady=5)

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

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

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

    def open_file(self):
        filename = self.entry_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:
            print("Ошибка: Файл не найден")

    def save_file(self):
        filename = self.entry_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)
        print("Информация: Файл успешно сохранен")

```

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

Результат работы программы:

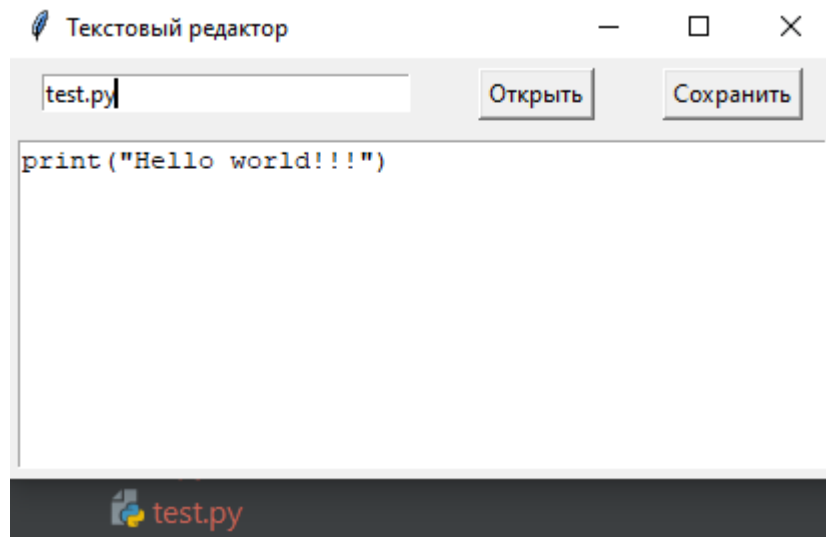


Рисунок 4. Результат работы программы к четвёртому заданию

Задание 5.

Код программы:

```
# -*- coding: utf-8 -*-

import tkinter as tk

def show_selected():
    selected_value.set(f"+ {selected_option.get()}")

root = tk.Tk()
root.title("Пример с Radiobutton")

selected_option = tk.StringVar()
selected_value = tk.StringVar()

radiobutton1 = tk.Radiobutton(root, text="Коля", width=8, height=3,
                              variable=selected_option, value=7588615151,
                              indicatoron=False, command=show_selected,
                              relief="flat")
radiobutton2 = tk.Radiobutton(root, text="Ваня", width=8, height=3,
                              variable=selected_option, value=7512458621,
                              indicatoron=False, command=show_selected,
                              relief="flat")
radiobutton3 = tk.Radiobutton(root, text="Вася", width=8, height=3,
                              variable=selected_option, value=7321458464,
                              indicatoron=False, command=show_selected,
                              relief="flat")

label = tk.Label(root, textvariable=selected_value, width=20, height=5)

radiobutton1.grid(row=0, column=0, sticky="w", padx=10, pady=2)
radiobutton2.grid(row=1, column=0, sticky="w", padx=10, pady=2)
radiobutton3.grid(row=2, column=0, sticky="w", padx=10, pady=2)
label.grid(row=0, column=1, rowspan=3, padx=10, pady=10)
```

```
root.mainloop()
```

Результат работы программы:

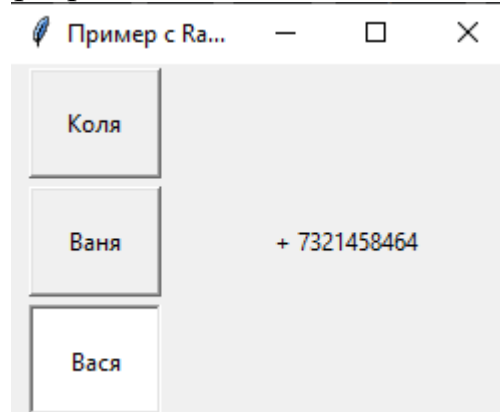


Рисунок 5. Результат работы программы к пятому заданию

Контрольные 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 по нажатию.

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