# МИНИCTEPCTBO 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) |

Ставрополь 2023

# Основы работы с tkinter

**Цель работы:** приобретение навыков построения графического интерфейса пользователя GUI с помощью пакетаTkinter 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:

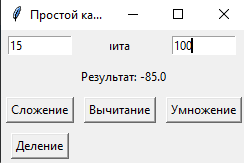


Рисунок 1. Результат работы программы к первому заданию

# 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:

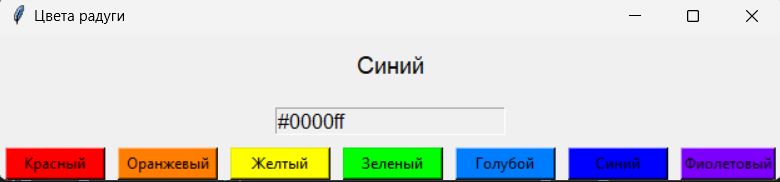


Рисунок 3. Результат работы программы к третей задачи

# 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:

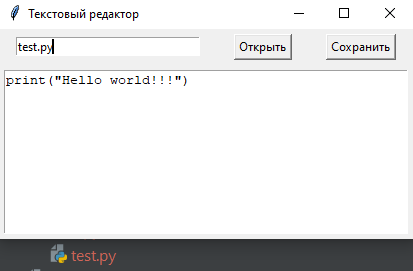


Рисунок 4. Результат работы программы к четвёртому заданию

# 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:

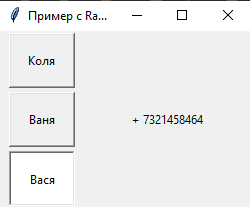


Рисунок 5. Результат работы программы к пятому заданию

# 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.

# 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.

# 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()).

# 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().

# 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.

# 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.

# 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.

# 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.

# 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).

# 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.

# Как осуществляется вставка виджетов в текстовое поле?

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

# 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.

# 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.

# 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 пакетаTkinter языка

программирования Python версии 3.x.