

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт перспективной инженерии
Департамент цифровых, робототехнических систем и электроники

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №7
дисциплины «Объектно-ориентированное программирование»

Выполнил:
Кожуховский Виктор Андреевич
3 курс, группа ИВТ-б-о-22-1,
09.03.01 «Информатика и
вычислительная техника»,
направленность (профиль)
«Программное обеспечение средств
вычислительной
техники и автоматизированных систем
», очная форма обучения

(подпись)

Проверил:
Воронкин Роман Александрович

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2024 г.

Тема: Основы работы с Tkinter

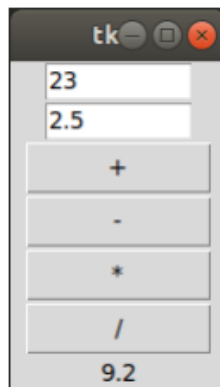
Цель: приобретение навыков построения графического интерфейса пользователя GUI с помощью пакета Tkinter языка программирования Python версии 3.x.

Порядок выполнения работы:

1. Изучил теоретический материал работы.
2. Создал общедоступный репозиторий на GitHub, в котором использована лицензия MIT и язык программирования Python.
3. Выполнил клонирование созданного репозитория.
4. Дополнил файл .gitignore необходимыми правилами для работы с IDE.
5. Организовал свой репозиторий в соответствии с моделью ветвления git-flow.
6. Создал проект в папке репозитория.
7. Проработал примеры лабораторной работы.
9. Разработайте программу по следующему описанию.

Задание 1

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".



```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

from tkinter import Tk, Entry, Button, Label

def calculate(operation: str) -> None:
    try:
        num1 = float(entry1.get())
        num2 = float(entry2.get())
        if operation == "+":
            result = num1 + num2
        elif operation == "-":
            result = num1 - num2
        elif operation == "*":
            result = num1 * num2
        elif operation == "/":
            result = num1 / num2
        label_result["text"] = result
    except ValueError:
        label_result["text"] = "Ошибка"

if __name__ == "__main__":
    root = Tk()

    entry1 = Entry(root, width=12)
    entry1.pack()

    entry2 = Entry(root, width=12)
    entry2.pack()

    button_add = Button(root, text="+", command=lambda: calculate("+"), width=10)
    button_add.pack()

    button_subtract = Button(root, text="-", command=lambda: calculate("-"), width=10)
    button_subtract.pack()

    button_multiply = Button(root, text="*", command=lambda: calculate("*"), width=10)
    button_multiply.pack()

    button_divide = Button(root, text="/", command=lambda: calculate("/"), width=10)
    button_divide.pack()

    label_result = Label(root)
    label_result.pack()

    root.mainloop()
```

Рисунок 1. Выполнение общего задания 1

Задание 2

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.

Примерно должно получиться так:



```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

from tkinter import Tk, Entry, Button, Label

def insert_color_code(color_code: str, color_name: str) -> None:
    entry.delete(0, "end")
    entry.insert(0, color_code)
    label_color["text"] = color_name

if __name__ == "__main__":
    root = Tk()

    label_color = Label(root, text="", width=12)
    label_color.pack()

    entry = Entry(root, width=13, bd=0, highlightthickness=0)
    entry.pack()

    colors = [
        ("Красный", "#ff0000"),
        ("Оранжевый", "#ff7d00"),
        ("Желтый", "#ffff00"),
        ("Зеленый", "#00ff00"),
        ("Голубой", "#007dff"),
        ("Синий", "#0000ff"),
        ("Фиолетовый", "#7d00ff"),
    ]

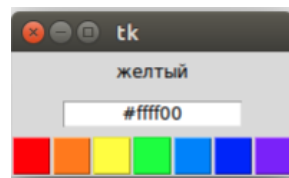
    for color_name, color_code in colors:
        button = Button(
            root,
            bg=color_code,
            command=lambda c=color_code, n=color_name: insert_color_code(c, n),
            width=10,
        )
        button.pack()

    root.mainloop()
```

Рисунок 2. Выполнение общего задания 2

Задание 3

Перепишите программу из пункта 8 так, чтобы интерфейс выглядел примерно следующим образом:



```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

from tkinter import Tk, Entry, Button, Label

def insert_color_code(color_code: str, color_name: str) -> None:
    entry.delete(0, "end")
    entry.insert(0, color_code)
    label_color["text"] = color_name

if __name__ == "__main__":
    root = Tk()

    label_color = Label(root, text="", width=12)
    label_color.pack()

    entry = Entry(root, width=13, bd=0, highlightthickness=0)
    entry.pack()

    colors = [
        ("Красный", "#ff0000"),
        ("Оранжевый", "#ff7d00"),
        ("Желтый", "#ffff00"),
        ("Зеленый", "#00ff00"),
        ("Голубой", "#007dff"),
        ("Синий", "#0000ff"),
        ("Фиолетовый", "#7d00ff"),
    ]

    for color_name, color_code in colors:
        button = Button(
            root,
            bg=color_code,
            command=lambda c=color_code, n=color_name: insert_color_code(c, n),
            width=2,
        )
        button.pack(side="left")

    root.mainloop()
```

Рисунок 3. Выполнение общего задания 3

Задание 4

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. Освежить знания о файлах можно из материала лабораторной работы 9

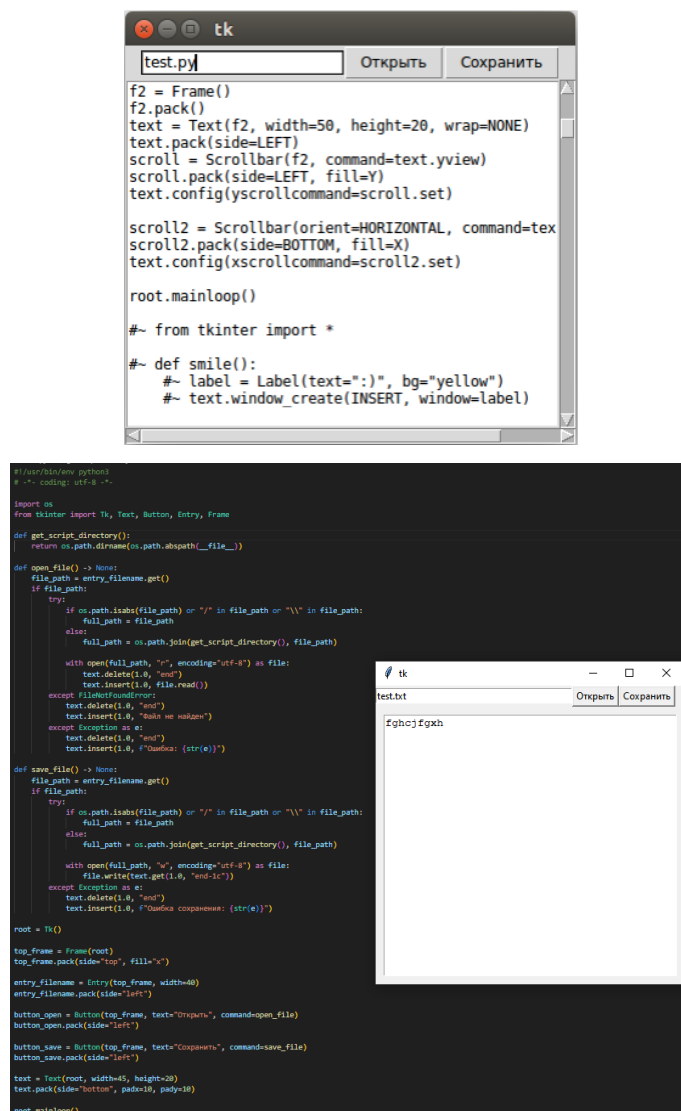


Рисунок 4. Выполнение общего задания 4

Задание 5

виджеты `Radiobutton` 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.

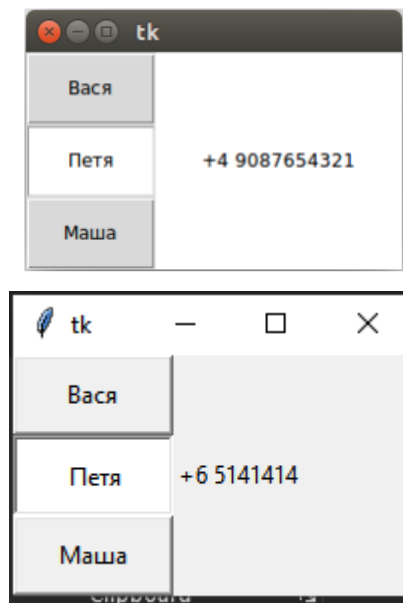


Рисунок 5. Выполнение общего задания 5

9. Зафиксировал сделанные изменения в репозитории.
10. Выполнил слияние ветки для разработки с веткой `master/main`.
11. Отправил сделанные изменения на сервер GitHub.

Ссылка: <https://github.com/Viktorkozh/OOP-7>

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.

2. Что такое Tkinter?

Tkinter – это пакет для Python, предназначенный для работы с библиотекой Tk. Библиотека Tk содержит компоненты графического интерфейса пользователя (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?

Цикл обработки событий – это метод `mainloop` экземпляра Tk, который запускает главный цикл обработки событий, что в том числе приводит к отображению главного окна со всеми "упакованными" на нем виджетами.

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

Экземпляр класса Tk создается от класса Tk модуля tkinter и представляет собой главное окно приложения, в котором располагаются все остальные виджеты.

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

Button – кнопка, которая выполняет действие при нажатии.

Label – метка, которая отображает текст в окне и служит в основном для информационных целей.

Entry – однострочное текстовое поле для ввода информации пользователем.

Text – многострочное текстовое поле для ввода информации пользователем.

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

Метод pack() используется для размещения виджетов в окне. Если к элементу интерфейса не применить какой-либо из менеджеров геометрии, то он не отобразится в окне.

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

Управление размещением виджетов с помощью метода pack() осуществляется путем передачи параметров, таких как side (сторона), который принимает одно из четырех значений - констант tkinter – TOP, BOTTOM, LEFT, RIGHT (верх, низ, лево, право).

9. Как осуществляется управление полосами прокрутки в виджете Text?

Управление полосами прокрутки в виджете Text осуществляется с помощью класса Scrollbar. Объект-скроллер связывается с виджетом, которому он требуется, и устанавливается опцией yscrollcommand.

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

Тэги при работе с виджетом Text предназначены для форматирования текста, то есть для придания его разным частям разного оформления.

11. Как осуществляется вставка виджетов в текстовое поле?

Вставка виджетов в текстовое поле осуществляется с помощью метода window_create, который позволяет вставлять другие виджеты в текстовое поле.

12. Для чего предназначены виджеты Radiobutton и Checkbutton?

Виджеты Radiobutton и Checkbutton предназначены для создания групповых переключателей и независимых флажков соответственно. Радиокнопки работают по принципу переключателей, а флажки могут быть установлены или сняты независимо друг от друга.

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

Переменные Tkinter – это специальные классы, такие как BooleanVar, IntVar, DoubleVar, StringVar, которые предназначены для хранения состояний виджетов и позволяют отслеживать их значения.

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

Связь переменных Tkinter с виджетами Radiobutton и Checkbutton осуществляется через свойство `variable`, которое устанавливается в одно и то же значение для всех радиокнопок одной группы, а для каждого флажка должна быть своя переменная Tkinter.

Вывод: приобрел навыки построения графического интерфейса пользователя GUI с помощью пакета Tkinter языка программирования Python версии 3.x.