

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образо-
вательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

**Кафедра инфокоммуникаций
«Основы работы с Tkinter»**

**Отчет по лабораторной работе № 4.7
по дисциплине «Объектно-ориентированное программирование»**

Выполнил студент группы ИВТ-б-о-21-1

Богадунов Василий Игоревич.

« » _____ 20__г.

Подпись студента _____

Работа защищена « » _____ 20__г.

Проверил Воронкин Р.А. _____
(подпись)

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

Порядок выполнения работы:


1. Создал общедоступный репозиторий на GitHub, в котором использована лицензия MIT и язык программирования Python.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (*).

Owner * Repository name *

 ItsMyLife1337 /

Great repository names are short and memorable. Need inspiration? How about [friendly-system](#) ?

Description (optional)



- ☒  **Public**
Anyone on the internet can see this repository. You choose who can commit.
- ☐  **Private**
You choose who can see and commit to this repository.

Рисунок 1 – Создание репозитория

2. Выполнение клонирования созданного репозитория.

```
c:\Users\Admin\Desktop\git>git clone https://github.com/ItsMyLife1337/OOP_4.3.git
Cloning into 'OOP_4.3'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (5/5), done.
c:\Users\Admin\Desktop\git>
```

Рисунок 2 – Клонирование репозитория

3. Организация репозитория в соответствии с моделью ветвления git-flow.

```

c:\Users\Admin\Desktop\git>cd /d c:\users\admin\desktop\git\OOP_4.3
c:\Users\Admin\Desktop\git\OOP_4.3>git flow init
Which branch should be used for bringing forth production releases?
- main
Branch name for production releases: [main]
Branch name for "next release" development: [develop]
How to name your supporting branch prefixes?
Feature branches? [feature/]
Bugfix branches? [bugfix/]
Release branches? [release/]
Hotfix branches? [hotfix/]
Support branches? [support/]
Version tag prefix? []
Hooks and filters directory? [C:/Users/Admin/Desktop/git/OOP_4.3/.git/hooks]
c:\Users\Admin\Desktop\git\OOP_4.3>

```

Рисунок 3 – Ветвление по модели git-flow

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".

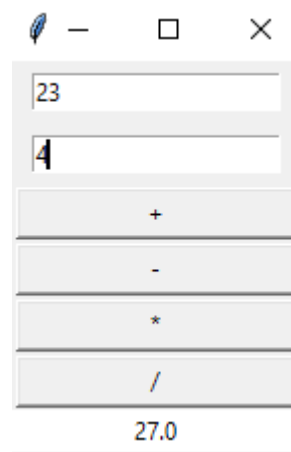


Рисунок 4 – Результат выполнения задания

Задание 2. 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.

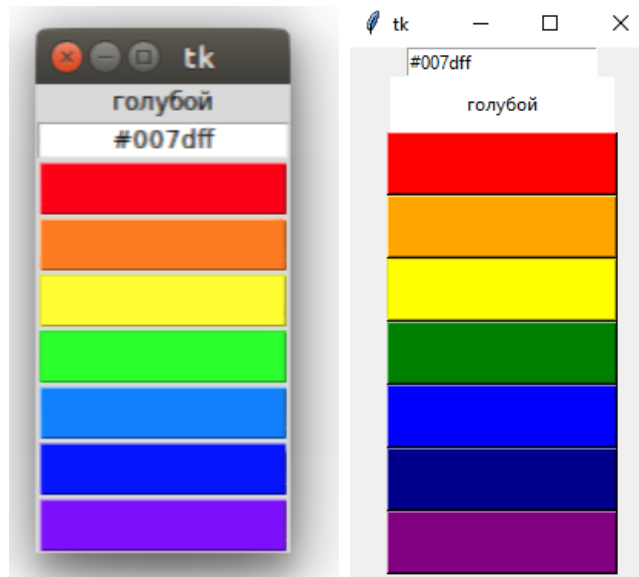


Рисунок 5 – Слева необходимый результат, справа полученный (написанный)

Задание 3. Решите задачу: перепишите программу из пункта 8 так, чтобы интерфейс выглядел примерно следующим образом:

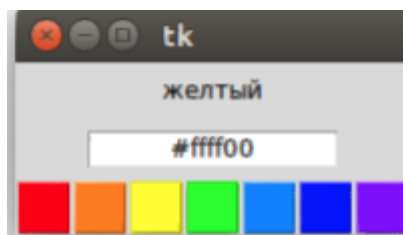


Рисунок 6 – Пример задания №3

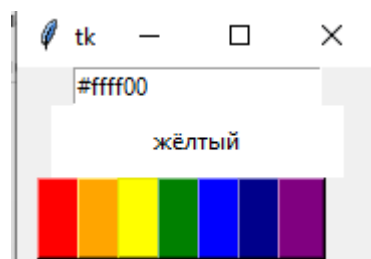


Рисунок 7 – Результат выполнения

Задание 4. 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. Для выполнения практической работы вам понадобится функция `open` языка Python и методы файловых объектов чтения и записи.

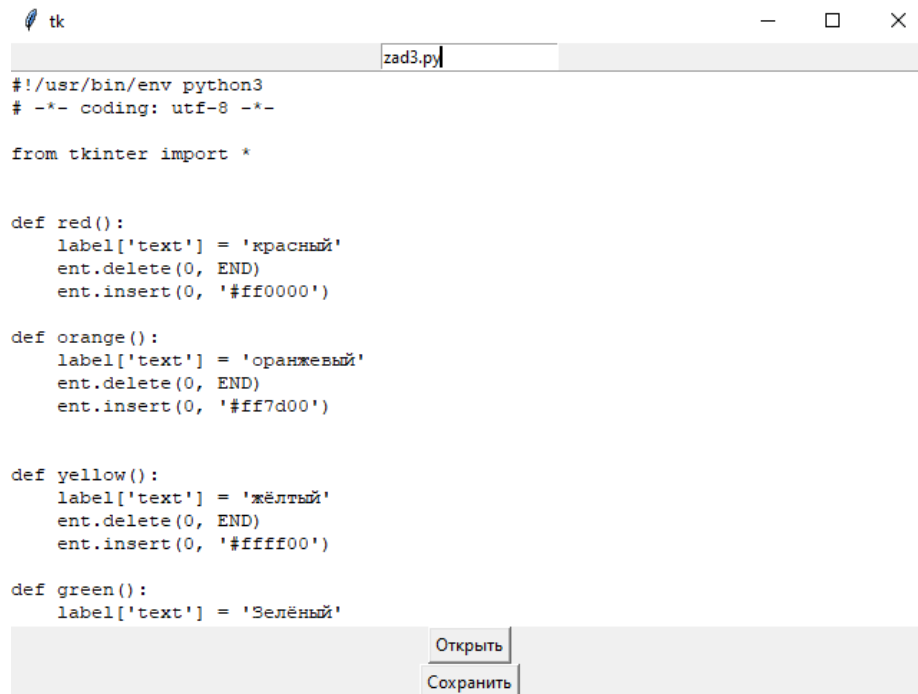


Рисунок 8 – Результат выполнения задания

Задание №5. Решите задачу: 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 9 – Пример и результат

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

Ответы на 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 при 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: 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 выбрать один из нескольких взаимо-исключающих 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.