

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»
Вариант 3

Выполнил:
Говоров Егор Юрьевич
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,
доцент департамента цифровых,
робототехнических систем и
электроники

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2024 г.

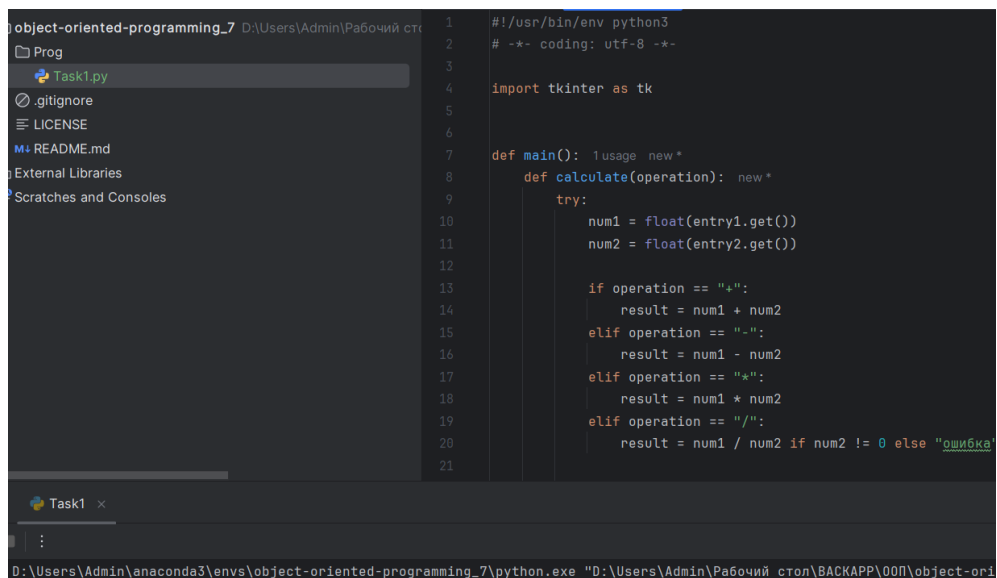
Тема: Основы работы с Tkinter

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

Ссылка на репозиторий: https://github.com/Artorias1469/object-oriented-programming_7.git

Ход работы:

Задача №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".



```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 import tkinter as tk
5
6
7 def main():
8     usage = new *
9     def calculate(operation):
10         new *
11         try:
12             num1 = float(entry1.get())
13             num2 = float(entry2.get())
14
15             if operation == "+":
16                 result = num1 + num2
17             elif operation == "-":
18                 result = num1 - num2
19             elif operation == "*":
20                 result = num1 * num2
21             elif operation == "/":
22                 result = num1 / num2 if num2 != 0 else "ошибка"
```

Рисунок 1. Код выполнения программы №1

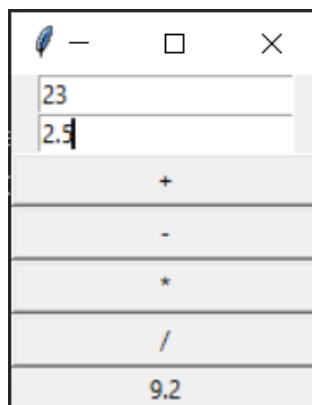
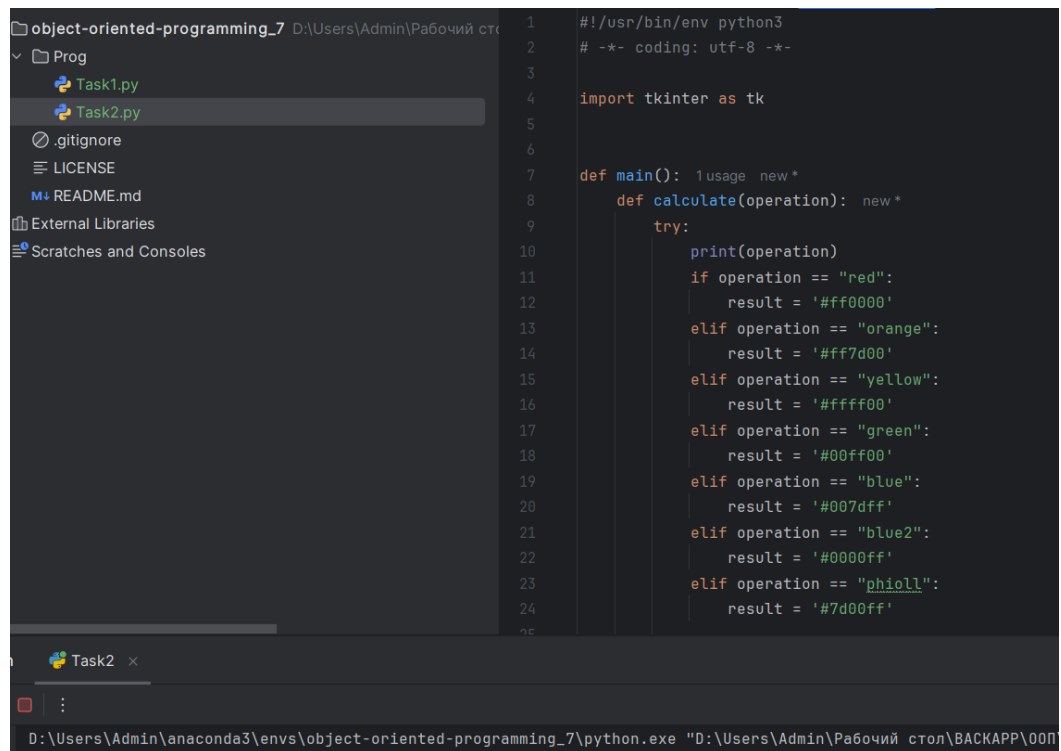


Рисунок 2. Результат выполнения работы №1

Задача №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.



```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  import tkinter as tk
5
6
7  def main():
8      def calculate(operation):
9          try:
10             print(operation)
11             if operation == "red":
12                 result = '#ff0000'
13             elif operation == "orange":
14                 result = '#ff7d00'
15             elif operation == "yellow":
16                 result = '#ffff00'
17             elif operation == "green":
18                 result = '#00ff00'
19             elif operation == "blue":
20                 result = '#007dff'
21             elif operation == "blue2":
22                 result = '#0000ff'
23             elif operation == "phiolet":
24                 result = '#7d00ff'
25
```

Рисунок 2. Код выполнения программы №2



Рисунок 3. Результат выполнения работы №2

Задача №3. Перепишите программу из пункта 8 так, чтобы интерфейс выглядел примерно следующим образом:

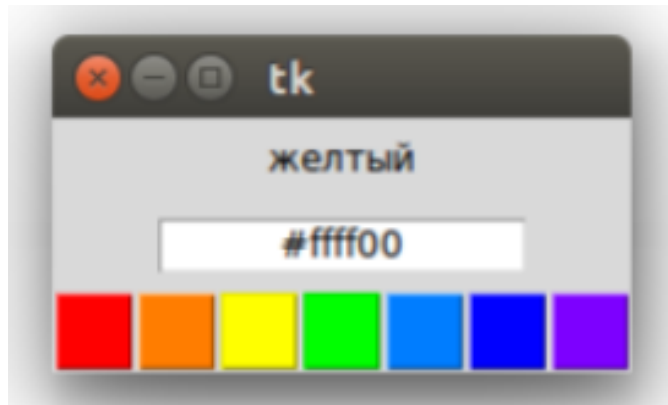


Рисунок 4. Образец выполнения

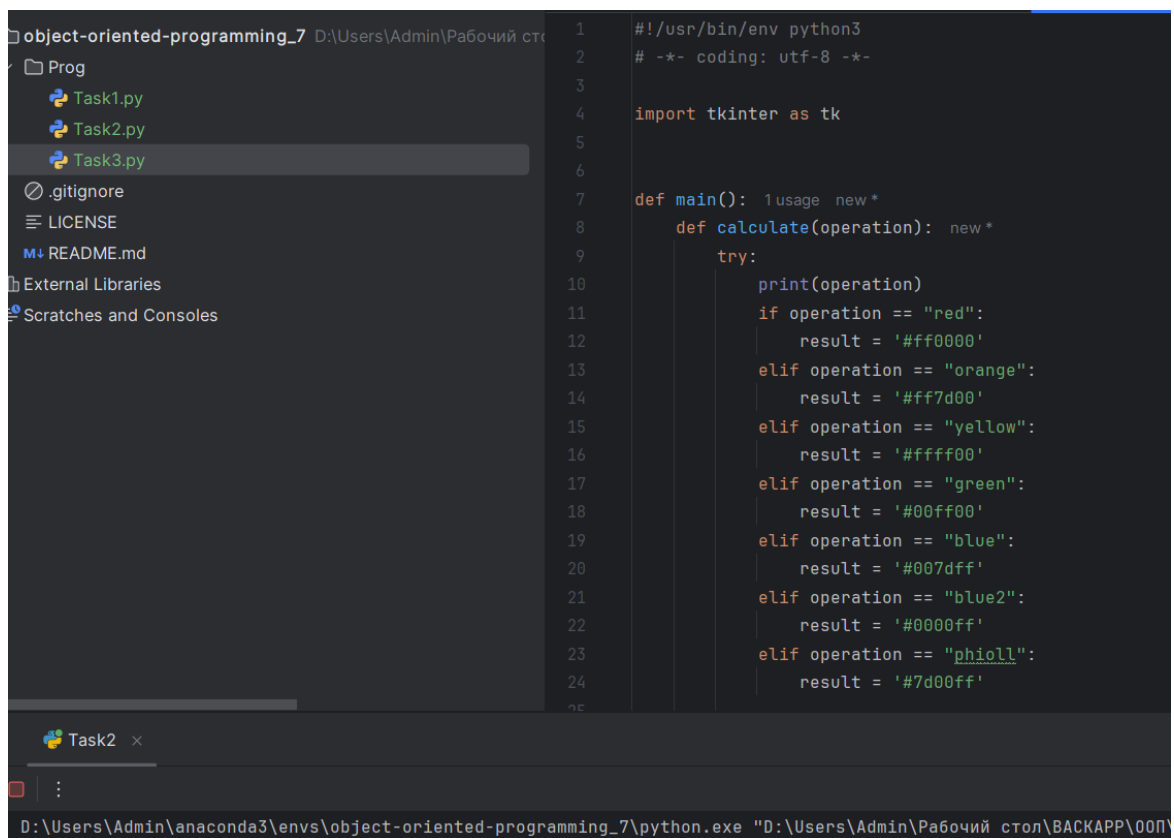


Рисунок 5. Код выполнения программы №3

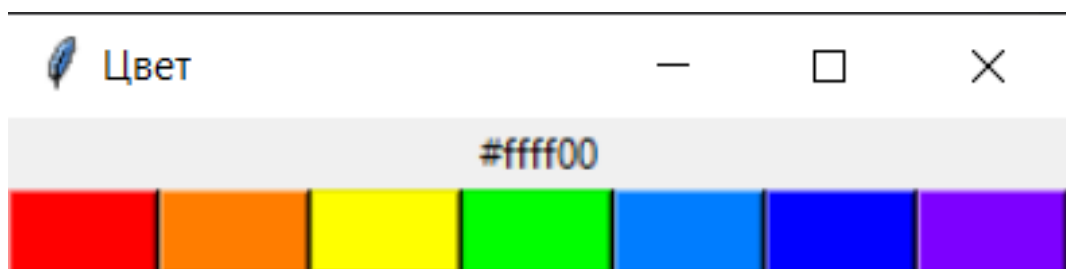


Рисунок 6. Результат выполнения работы №3

Задача №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.

```

1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  import tkinter as tk
5  from tkinter import filedialog, messagebox
6
7
8  def main():
9      usage = new *
10     def open_file():
11         filename = entry_filename.get()
12         try:
13             with open(filename, 'r', encoding='utf-8') as file:
14                 content = file.read()
15                 text_area.delete(index1: 1.0, tk.END) # Очистка текстового поля
16                 text_area.insert(tk.END, content) # Вставка содержимого файла
17         except FileNotFoundError:
18             messagebox.showerror(title="Ошибка", message="Файл не найден.")
19         except Exception as e:
20             messagebox.showerror(title="Ошибка", str(e))
21
22     def save_file():
23         filename = entry_filename.get()
24         try:
25             with open(filename, 'w', encoding='utf-8') as file:
26                 content = text_area.get(index1: 1.0, tk.END) # Вставка текста

```

Рисунок 7. Код выполнения программы №4

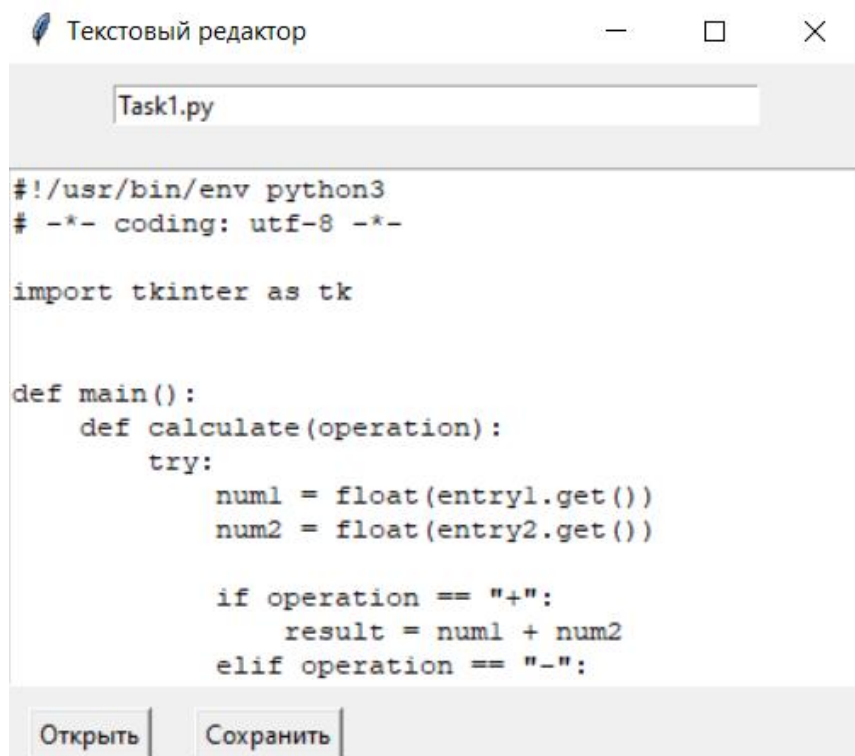


Рисунок 8. Результат выполнения работы №4

Задача №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.

```
name = selected_option.get()
if name == 'Егор':
    label.config(text=f"+ 7 (918) 794-98-77")
elif name == 'Дима':
    label.config(text=f"+ 7 (918) 794-88-77")
elif name == 'Андрей':
    label.config(text=f"+ 7 (918) 794-90-77")
```

Рисунок 9. Код выполнения программы №5 и реализация команды по изменению label

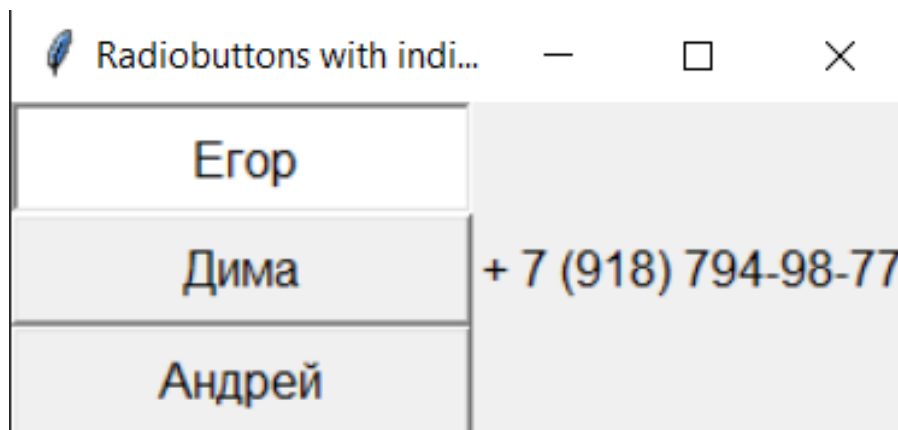


Рисунок 10. Результат выполнения работы №5

Ответы на контрольные вопросы:

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

В стандартной библиотеке Python для создания графического интерфейса (GUI) можно использовать: `tkinter` — наиболее распространённая библиотека для создания GUI, `turtle` — простая графическая библиотека для обучения и рисования, `IDLE` — встроенная среда разработки Python, которая тоже построена с использованием `Tkinter`.

Для более сложных интерфейсов часто используют сторонние библиотеки, такие как `PyQt`, `PySide`, `Kivy` или `wxPython`.

2. Что такое `Tkinter`?

`Tkinter` — это стандартная библиотека Python для создания графического интерфейса пользователя. Она предоставляет обёртку над инструментарием `Tk`, который является кроссплатформенным GUI-фреймворком. `Tkinter` поддерживает создание окон, кнопок, текстовых полей, меток, списков, радиокнопок и других элементов интерфейса.

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

1) Импортировать модуль `Tkinter`: `import tkinter as tk`

2) Создать главное окно: `root = tk.Tk()`

3) Добавить виджеты (элементы интерфейса): Например, кнопки, метки, текстовые поля.

4) Разместить виджеты: Использовать методы `pack()`, `grid()` или `place()` для расположения элементов в окне.

5) Запустить цикл обработки событий: `root.mainloop()`

4. Что такое цикл обработки событий?

Цикл обработки событий — это процесс, при котором приложение ожидает действий пользователя (например, нажатия кнопки, ввода текста) и реагирует на них. В Tkinter этот цикл запускается с помощью метода `mainloop()`. Он обрабатывает все события, поступающие в программу.

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

Экземпляр класса Tk представляет главное окно приложения. Он создаёт базовое окно, в котором размещаются все виджеты. Это окно управляет жизненным циклом GUI, включая отрисовку, обновление и закрытие.

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

Button — кнопка, на которую можно нажать для выполнения действия. Label — текстовая метка, предназначенная для отображения текста или изображений. Entry — однострочное текстовое поле для ввода данных. Text — многострочное текстовое поле для ввода и редактирования текста.

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

Метод `pack()` используется для автоматического размещения виджетов в контейнере (например, главном окне или рамке). Он упрощает компоновку

элементов, размещая их по порядку в зависимости от направления (по вертикали или горизонтали).

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

Параметры управления: `side` — определяет сторону контейнера, к которой примыкает виджет (`top`, `bottom`, `left`, `right`). `fill` — позволяет виджету заполнять доступное пространство (`x`, `y`, `both`). `expand` — указывает, должен ли виджет расширяться при увеличении размера контейнера. `padx` и `pady` — добавляют отступы по горизонтали и вертикали.

9. Как осуществляется управление полосами прокрутки в виджете `Text`?

Для управления полосами прокрутки необходимо:

- 1) Создать объект `Scrollbar`.
- 2) Связать его с виджетом `Text` с помощью параметров `yscrollcommand` (для вертикальной прокрутки) или `xscrollcommand` (для горизонтальной).
- 3) Связать `Scrollbar` с методом прокрутки `Text`.

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

Тэги позволяют применять различные стили или действия к определённым частям текста в виджете `Text`. С их помощью можно: Изменять цвет, шрифт или фон текста, реализовывать обработчики событий (например, щелчок по выделенному тексту), удобно управлять форматированием.

11. Как осуществляется вставка виджетов в текстовое поле?

Для вставки виджета в текстовое поле используется метод `window_create`:

```
button = tk.Button(root, text="Кнопка") text.window_create("end",  
window=button)
```

12. Для чего предназначены виджеты Radiobutton и Checkbutton?

Radiobutton — предоставляет возможность выбрать одну из нескольких опций (взаимоисключающий выбор). Checkbutton — позволяет выбрать несколько опций одновременно (независимые флажки).

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

Переменные Tkinter (StringVar, IntVar, DoubleVar, BooleanVar) — это специальные объекты, которые связывают значения виджетов с программным кодом. Изменение переменной автоматически обновляет связанный виджет, и наоборот.

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

Radiobutton: Связывается с переменной с помощью параметра variable.

Каждая кнопка имеет уникальное значение, задаваемое через value.

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