

Практическое занятие №7

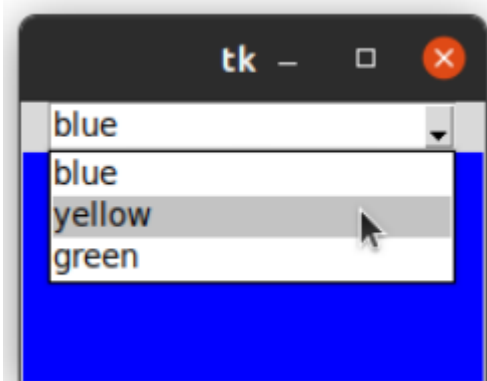
Создание графического интерфейса с помощью Tkinter

Порядок выполнения

I. Ответьте на вопросы теста

II. Выполните задания:

1. Напишите программу, состоящую из выпадающего списка (Combobox) и холста. В списке перечисляются цвета. При выборе в списке того или иного значения цвет холста должен изменяться на соответствующий.



Для справки:

- Модуль `ttk` содержит классы более стилизованных и современных виджетов
- Метод `pack` (упаковщик) имеется у всех виджетов и используется для их размещения в окне. Если ему не передать аргументов, то виджеты будут располагаться вертикально, друг под другом.
- В `tkinter` с помощью метода `bind` между собой связываются виджет, событие и действие

Алгоритм:

1. Подключаем необходимые модули
2. Создаем окно и настраиваем его размеры, заголовок
3. Создаем список `colors` значений комбобокса
4. Создаем комбобокс и размещаем его в окне
5. Список значений передается в `Combobox` через опцию `values`:

```
combo = Combobox(values=colors)
```
6. Метод `current` экземпляра `Combobox` позволяет указать значение, которое будет выбрано в списке изначально. Метод принимает индекс элемента из списка значений:

```
combo.current(0) - выбран самый верхний
```

7. Создаем холст как экземпляр класса Canvas . Задаем ширину, высоту, цвет – опять же первый из списка:

```
canvas = Canvas(width=200, height=100, bg=colors[0])
```

Размещаем холст в окне

8. Применяем метод bind для связи объекта, события и действия. Событие смены значения в выпадающем списке – '<ComboboxSelected>'.
Действие – изменение цвета канвы. Метод get экземпляра Combobox возвращает выбранный на данный момент элемент списка.

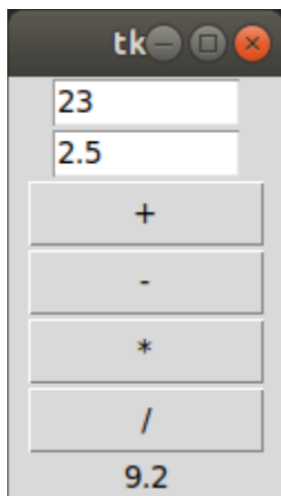
Действие – изменение цвета канвы. Метод get экземпляра Combobox возвращает выбранный на данный момент элемент списка.

```
combo.bind('<ComboboxSelected>', lambda  
event: canvas.config(bg=combo.get()))
```

9. Задаем бесконечный цикл

Напишите комментарии к каждому блоку программы, соответствующему шагу алгоритма. Запустите программу, добейтесь ее правильной работы. Добавьте в список еще два цвета. Нужно ли при этом что-то еще менять в программе?

2. Напишите простейший калькулятор, состоящий из двух текстовых полей, в которые пользователь вводит числа, и четырех кнопок «+», «-», «*», «/». Результат вычисления должен отображаться в надписи. Если арифметическое действие выполнить невозможно (например, если введены буквы, а не цифры), то в надписи должно появиться сообщение «ошибка».



Для справки:

- Для чтения значений из поля ввода используется метод `get()`. Помните, что поле ввода содержит строку!
- Для обработки исключений применяется конструкция `try-except`. В блоке `try` выполняются инструкции, которые могут породить исключение, а в блоке `except` эти исключения перехватываются. Например:

```
try:  
    k = 1/0  
except ZeroDivizionError:
```

k = 0

Программа:

```
# заполнить - подключение библиотеки
class MathButton:
    def __init__(self, action):
        self.but = Button(text=action, width=10,
            command=self.math)
        self.operator = action
        self.but.pack()
    def math(self):
        try:
            n1 = # заполнить
            n2 = # заполнить
        except ValueError:
            lab_result['text'] = # заполнить
        else:
            if self.operator == '+':
                lab_result['text'] = round(n1 + n2, 2)
            # заполнить для операций '-' и '*'
            elif self.operator == '/':
                if n2 != 0:
                    lab_result['text'] = # заполнить
                else:
                    lab_result['text'] = 'ZeroDivisionError'

# Заполнить - создание и настройка окна
# Заполнить - создание и размещение полей ввода
but_summa = MathButton('+')
but_diff = # заполнить
but_prod = # заполнить
but_div = # заполнить
# Заполнить - создание и размещение надписи lab_result
# заполнить - бесконечный цикл
```

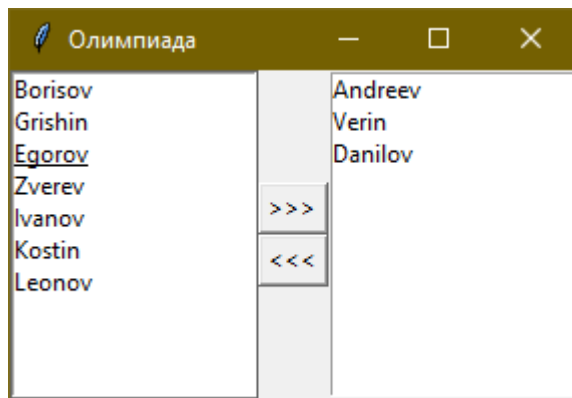
Заполните пропуски в коде программы.

Введите ее, отладьте, добейтесь правильной работы.

Проверьте, как программа реагирует на исключения.

Увеличьте размер шрифта полей ввода, кнопок, надписи до 15 пунктов.

3. Напишите приложение, работающее с двумя списками `Listbox`. В первом будет, например, список курсантов, заданный программно. Второй изначально пуст, пусть это будет список участников олимпиады. При клике на одну кнопку курсант должен переходить из одного списка в другой. При клике на вторую кнопку – возвращаться (человек передумал участвовать). Предусмотрите возможность множественного выбора элементов списка и их перемещения.



Программа:

```
# заполнить - подключение библиотеки
def choice(first, second):
    select = first.curselection()
    for i in select:
        second.insert(END, first.get(i))
    select = list(select)
    select.reverse()
    for i in select:
        first.delete(i)

# заполнить - создание и настройку окна
f = Frame()
f.pack(side=LEFT)

# заполнить - создание списка курсантов
# Заполнить - создание двух списков, настройка множественного
# выбора, их размещение
Button(f, text=">>>", command=lambda i=group, j=olimp: choice
(i, j)).pack()

# заполнить - создание второй кнопки и ее размещение
# заполнить - бесконечный цикл
```

Прокомментируйте каждую строчку метода `choice`.

Заполните пропуски в программе.

Выполните программу.

Измените ее так, чтобы курсанты не удалялись из левого списка, но удалялись из правого списка при отказе участвовать.

- Создайте приложение, предъявляющее пользователю один вопрос теста с множественным выбором и выдающее реакцию «верно/неверно».

Текст вопроса и варианты ответов придумайте самостоятельно.

Для справки:

- Множественный выбор обеспечивают виджеты `checkbutton` (флажок)
- Флажок реагирует на команду
- Выбран флажок или нет, проверяет метод `get()`, который возвращает 1 или 0.

заполнить - подключение библиотеки

определить функцию select, проверяющую правильность выбора и выдающую реакцию «верно/неверно»

заполнить - создание окна, его настройка

создать две надписи - для вопроса и для реакции, разместить их

создать три флажка - три варианта ответа и разместить их

бесконечный цикл

Совет:

Пишите программу поэтапно:

Сначала настройте главное окно и расположите виджеты

Затем напишите метод(функцию), проверяющую выбор

Добавьте флажкам реакцию на выбор