Модуль *tkinter*. Создание и настройка окна

Tkinter – это пакет для *Питона*, предназначенный для работы с библиотекой *Tk* (обычно ее называют *ток*).

Библиотека *Tk* содержит компоненты графического интерфейса пользователя (*graphical user interface – GUI*). Эта библиотека написана на скриптовом языке *Tcl (Tool Command Language* или *тикл)*.

GUI – это окна, кнопки, текстовые поля для ввода, скроллеры, списки, радиокнопки, флажки и другие элементы управления. Все эти элементы интерфейса в *Tkinter* называются ***виджетами*** *(widgets).*

Импортируем модуль *tkinter*:

```
from tkinter import *
```

Основным компонентом программ с графическим интерфейсом является окно. Затем в окно добавляются остальные компоненты GUI. В *tkinter* главное окно программы представлено классом *Tk.*

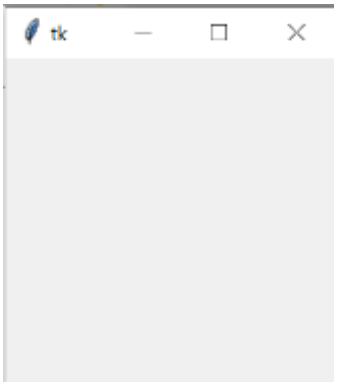
Создадим экземпляр класса *Tk* – объект главного окна. Обычно ему дают имя *root* (корневой) или *window* (окно), но мы для краткости назовем его *tk*. С помощью этой переменной мы будем управлять в дальнейшем атрибутами этого окна.

```
tk = Tk()
```

Для отображения окна надо вызвать для него метод *mainloop*(), который запускает цикл обработки событий окна для взаимодействия с пользователем. Окно не будет закрыто, пока сам пользователь не сделает это.

```
tk.mainloop()
```

Если вы запустите эту программу, то увидите следующее окно



Вы можете изменить размеры окна, распахнуть его на весь экран, свернуть в кнопку на панели задач или закрыть. Заголовком окна будет по умолчанию строка *“tk”*

Методы *minsize* и *maxsize* позволяют задать минимальные и максимальные размеры при изменении пользователем размеров окна. Первый параметр – количество пикселов по ширине, второй – по высоте.

```
tk.minsize(200,150)
tk.maxsize(500,300)
```

Чтобы задать свой заголовок окна, используем метод *title*. Его параметр – строка заголовка. Перед строкой с вызовом метода *mainloop*() добавьте следующую строку

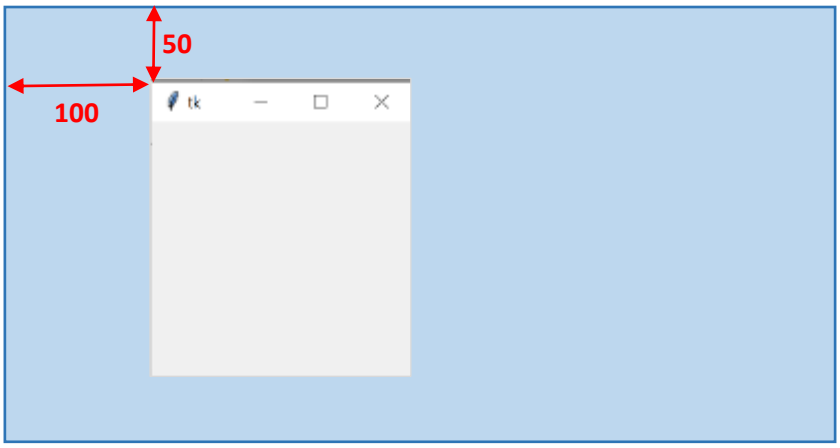
```
tk.title('Основное окно программы')
```

Для установки произвольных размеров окна используется метод *geometry*. Первое число в строке – ширина, второе – высота.

```
tk.geometry('800x400')
```

По умолчанию окно позиционируется в верхний левый угол экрана с некоторым смещением. Мы можем изменить его положение, задав после размеров окна смещение верхнего левого угла окна вправо и вниз от верхнего левого окна экрана.

```
tk.geometry('800x400+0+0') # смещение и по высоте, и по ширине равно 0
tk.geometry('800x400+100+50') # на 100px правее и на 50px ниже
```



Если вы зададите слишком большое смещение (например, больше ширины или высота экрана монитора), окно может исчезнуть из вашей видимости – нахождение окна в пределах экрана не контролируется. Но увидеть это окно среди открытых окон вы сможете стандартными способами *Windows* – кнопка на панели задач или переключаясь между окнами клавишами <*Shift-Tab*> или <*Alt-Esc*>

С помощью метода *resizable* можем запретить пользователю менять размеры окна. Мы установим запрет менять размер окна и по ширине *(width)* и по высоте *(height)*

```
tk.resizable(width=False, height=False)
```

Метод *configure* или *config* позволяет менять свойства виджетов. Поменяем цвет фона окна с помощью свойства *bg* или *background*. Значение свойства – цвет фона окна – можно задать либо названием цвета, либо шестнадцатиричным кодом числа. Более подробно о задании цвета в *Питоне* мы поговорим в следующий раз.

```
tk.configure(bg='#33cccc')
tk.config(background='yellow')
```

Текст полученной программы

```
from tkinter import *                          # подключаем модуль
tkinter
tk = Tk()                                      # создаем объект
главного окна
tk.title('Основное окно программы')            # задаем заголовок окна
tk.geometry("800x400+100+50")                  # задаем размеры и
смещение окна
tk.resizable(width=False, height=False) # запрещаем менять
размеры окна
tk.config(bg='#999999')                        # задаем цвет фона окна
tk.mainloop()
```

Дополнительная информация

Установка иконки

Перед заголовком окна отображается иконка. По умолчанию это изображение пера. С помощью метода *iconbitmap* можно задать любую другую иконку.

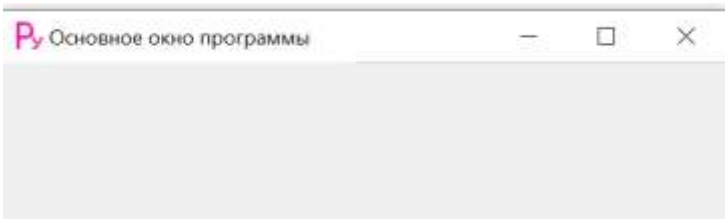
Предварительно этот файл с расширением *ico* надо создать. В этом вам помогут онлайн-редакторы и онлайн-конвертеры. Вот некоторые из них:

- https://bestwebservice.ru/ico_converter/
- https://iconsflow.com/ru
- https://online-converting.ru/image/convert-to-ico/

В этой статье приведен обзор подобных онлайн-средств: https://lumpics.ru/how-to-create-an-icon-ico-online/

Обычно файлу с иконкой приложения дают имя *favicon* (от англ. *favorite icon*). В указанном примере созданный файл иконки *favicon.ico* должен располагаться в той же папке, что и файл приложения. А вообще можно в значении параметра *default* указать полное имя иконки (вместе с путем).

```
root.iconbitmap(default="favicon.ico")
```



Изменение атрибутов окна

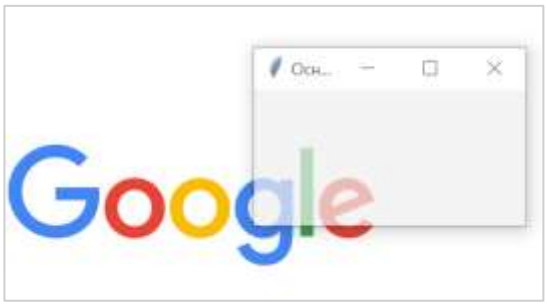
С помощью специального метода *attributes* можно установать отдельные атрибуты окна, для которых нет специальных методов. Первый параметр метода – название атрибута, которое предваряется дефисом. Второй параметр – значение этого атрибута.

Пример 1. Растяжение окна на весь экран (полноэкранный режим)

```
tk.attributes("-fullscreen", True)
```

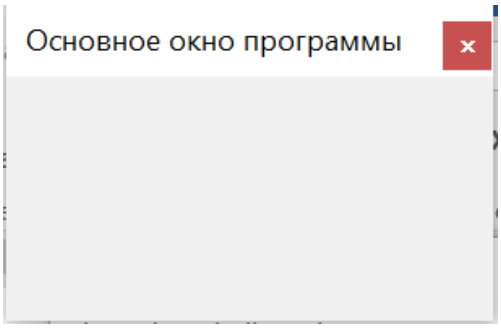
Пример 2. Установка прозрачности. Значение атрибута – вещественное число от 0 (полная прозрачность) до 1 (полная непрозрачность).

```
tk.attributes("-alpha", 0.5)
```



Пример 3. Отключение верхней панели окна (за исключением заголовка и кнопки закрытия окна).

```
tk.attributes("-toolwindow", True)
```



Позиционирование окна приложения в центре экрана

Методы *winfo_screenheight* и *winfo_screenwidth* позволяют получить размеры монитора.

```
monitor_w = tk.winfo_screenwidth()
monitor_h = tk.winfo_screenheight()
```

Посчитаем смещения окна по ширине и высоте для центровки окна:

смещение по ширине = (ширина монитора – ширина окна) / 2
смещение по высоте = (высота монитора – высота окна) / 2

Будем использовать целочисленное деление.

```
diff_w = (monitor_w - 800) // 2
diff_h = (monitor_h - 400) // 2
```

Так как параметр метода *geometry* должен быть строкового типа, а значения смещений целочисленные, то преобразуем целые числа в строку функцией *str*.

Операция + применительно к строкам называется ***операцией конкатенации (склейки)*** и позволяет получить строку, склеенную из разных строк.

```
tk.geometry('800x400+' + str(diff_w) + '+' + str(diff_h))
```

Можно было написать немного иначе, задав ширину и высоту окна тоже как целочисленные переменные:

```
w = 800
h = 400
diff_w = (monitor_w - w) // 2
diff_h = (monitor_h - h) // 2
tk.geometry(str(w)+'x'+str(h)+'+'+str(diff_w)+'+'+str(diff_h
))
```