

## Занятие 25. Элементы интерфейса библиотеки Tkinter

### План занятия

1.	Библиотека Tkinter языка Python.....	2
1.1.	Назначение Tkinter .....	2
1.2.	Создание оконного интерфейса.....	2
1.3.	Создание окна.....	4

# 1. БИБЛИОТЕКА TKINTER ЯЗЫКА PYTHON

## 1.1. Назначение Tkinter

Tkinter (от англ. tk interface) - это графическая библиотека, позволяющая создавать программы с оконным интерфейсом. Эта библиотека является интерфейсом к популярному языку программирования и инструменту создания графических приложений tcl/tk. Tkinter, как и tcl/tk, является кроссплатформенной библиотекой и может быть использована в большинстве распространённых операционных систем (Windows, Linux, Mac OS X и др.). Использование Tkinter позволяет создать такое окно:

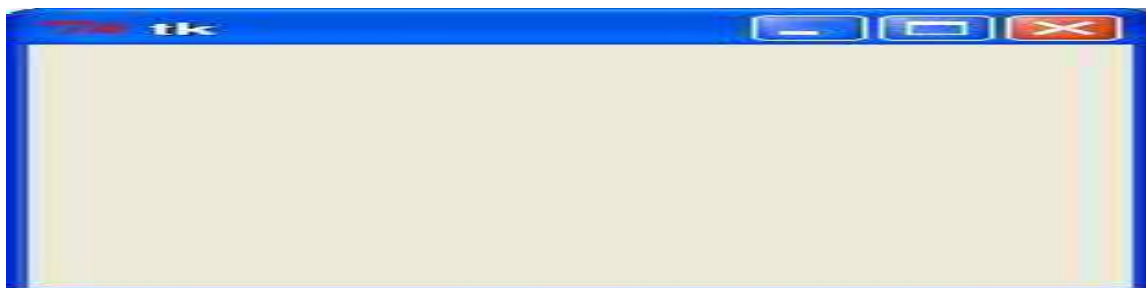


Рис. 1. Окно, созданное Tkinter

Справочная информация по Tkinter приводится, например, по адресу:  
[http://www.w3ii.com/ru/python/python\\_gui\\_programming.html](http://www.w3ii.com/ru/python/python_gui_programming.html).

## 1.2. Создание оконного интерфейса

Оконный интерфейс может представлять совокупность окон (фреймов) с размещенными на них различными элементами управления, которые называются виджетами (англ. widget).

Ниже перечислены важнейшие из них:

**Button** – это виджет, представляющий кнопку;

**Label** - это виджет, предназначенный для отображения какой-либо надписи без возможности редактирования пользователем;

**Entry** - это виджет, позволяющий пользователю ввести одну строку текста. Имеет дополнительное свойство `bd` (сокращённо от `borderwidth`), позволяющее регулировать ширину границы;

**Text** - это виджет, который позволяет пользователю ввести любое количество текста. Имеет дополнительное свойство `wrap`, отвечающее за перенос (чтобы, например, переносить по словам, нужно использовать значение `WORD`);

**Listbox** - это виджет, который представляет собой список, из элементов которого пользователь может выбирать один или несколько пунктов. Имеет дополнительное свойство `selectmode`, которое, при значении `SINGLE`, позволяет пользователю выбрать только один элемент списка, а при значении `EXTENDED` - любое количество;

**Frame** - виджет (рамка) предназначен для организации виджетов внутри окна;

**Checkbutton** - это виджет, который позволяет отметить „галочкой“ определенный пункт в окне. При использовании нескольких пунктов нужно каждому присвоить свою переменную;

**Radiobutton** – этот виджет выполняет функцию, схожую с функцией виджета `Checkbutton`. Разница в том, что в виджете `Radiobutton` пользователь может выбрать лишь один из пунктов;

**Scale** (шкала) - это виджет, позволяющий выбрать какое-либо значение из заданного диапазона;

**Scrollbar** - этот виджет даёт возможность пользователю "прокрутить" другой виджет (например текстовое поле) и часто бывает полезен. Использование этого виджета достаточно нетривиально. Необходимо сделать две привязки: `command` полосы прокрутки привязываем к методу `xview/yview` виджета, а `xscrollcommand/yscrollcommand` виджета привязываем к методу `set` полосы прокрутки.

**Progressbar** - виджет отображает уровень загрузки;

**Combobox** - виджет `Combobox` предназначен для отображения списка значений, их выбора или изменения пользователем. В версии `tk` ему подобен виджет `Listbox`. Разница заключается в том, что `Combobox` имеет возможность сворачиваться подобно свитку, а `Listbox` будет отображаться всегда открытым.

### 1.3. Создание окна

Для создания окон интерфейса предварительно необходимо выполнить импорт библиотеки:

`from Tkinter import *` или `import Tkinter` (версия python 2.7 и ниже);

`from tkinter import *` или `import tkinter` (версия python 3.0 и выше);

Создадим проект TkinterWindow для иллюстрации создания окна:

**File->New Project – задать TkinterWindow.**

В браузере проектов выделим проект ..\TkinterWindow, затем, используя правую клавишу мыши, создадим файл TkinterWindow.py.

Откроется закладка TkinterWindow текстового редактора.

Внесем следующий текст:

```
import Tkinter
top = Tkinter.Tk()
# Code to add widgets will go here...
top.mainloop()
```

Здесь:

**import Tkinter** – импортирует библиотеку классов объектов Tkinter.

Tk является базовым классом любого Tkinter приложения. При создании объекта этого класса запускается интерпретатор tcl/tk и создается базовое окно приложения.

Предложение `top = Tkinter.Tk()` создает экземпляр объекта класса Tk с именем top.

Предложение `top.mainloop()` обозначает вызов метода `mainloop()` экземпляром объекта top. Метод `mainloop()` создает окно запуском главного цикла обработки событий.

Примечание. Tkinter является событийно-ориентированной библиотекой. В приложениях такого типа имеется главный цикл обработки событий. В Tkinter такой цикл запускается методом `mainloop`. Для явного выхода из интерпретатора и завершения цикла обработки событий используется метод `quit`.

Запустим программу:

Run->Run TkinterWindow->Run TkinterWindow.

В результате выполнения программы будет выведено простое окно. Тем, не менее, обратите внимание на то, что размеры окна и его местоположение можно произвольно менять. Кроме того, возможно задать полноэкранный режим (на весь экран).

Закрытие окна приведет к закрытию программы.

Закроем проект TkinterWindow.