Введение в tkinter

tkinter – библиотека, в которую включены виджеты для языка программирования Python .

Этапы получения программы с GUI

Импорт библиотеки

Создание главного окна

Создание виджет

Установка их свойств

Определение событий

Определение обработчиков событий

Расположение виджет на главном окне

Отображение главного окна

Импорт модуля tkinter

Способы импортирования:

1. import tkinter
2. from tkinter import *

Предпочтительно пользоваться вторым способом, т. к. это позволит не указывать каждый раз имя модуля при обращении к объектам, которые в нем содержатся.

Первая строка программы:

```
from tkinter import *
```

Создание главного окна

Любое пользовательское приложение заключено в окно, которое можно назвать главным, т.к. в нем располагаются все остальные виджеты. Объект окна верхнего уровня создается при обращении к классу Tk модуля tkinter.

Переменную связанную с объектом-окном принято называть root (хотя понятно, что можно назвать как угодно, но так уж принято).

Вторая строчка кода:

```
root = Tk()
```

Создание виджет

Кнопка создается при обращении к классу Button модуля tkinter. Объект кнопка связывается с какой-нибудь переменной. У класса Button (как и всех остальных классов, за исключением Tk) есть обязательный параметр — объект, которому кнопка принадлежит (кнопка не может "быть ничейной"). Единственное окно (root), является аргументом, передаваемым в класс при создании объекта-кнопки:

```
but = Button(root)
```

Установка свойств виджет

У кнопки много свойств: размер, цвет фона и надписи и др. Установим всего одно свойство — текст надписи (text):

```
but["text"] = "Печать"
```

Определение событий и их обработчиков

Предположим, что задача кнопки вывести какое-нибудь сообщение в поток вывода, используя функцию print. Делать она это будет при нажатии на нее левой кнопкой мыши.

Алгоритм (действия) оформляют в виде функции, а затем вызывают, когда они понадобятся.

```
def printer(event):
print ("Как всегда очередной 'Hello World!'")
```

*Параметр event – это какое-либо событие

Событие нажатия левой кнопкой мыши выглядит так: <Button-1>.

Требуется связать это событие с обработчиком (функцией printer).

Для связи предназначен метод bind. Синтаксис связывания события с обработчиком выглядит так:

```
but.bind("<Button-1>",printer)
```

Размещение виджет

Влюбом приложении виджеты не разбросаны по окну как попало, а хорошо организованы, интерфейс продуман до мелочей и обычно подчинен определенным стандартам. Для того чтобы отобразить кнопку в окне используют метод pack.

```
but.pack()
```

Если не вставить эту строчку кода, то кнопка в окне так и не появится, хотя она есть в программе.

Отображение главного окна

mainloop – метод вызова главного окна

```
but.pack()
```

Данная строчка кода должна быть всегда в конце скрипта

Код программы

```
from tkinter import *

def printer(event):
    print ("Как всегда очередной 'Hello World!'")

root = Tk()
but = Button(root)
but["text"] = "Печать" but.bind("<Button-1>",printer)

but.pack()
root.mainloop()
```

При программировании графического интерфейса пользователя более эффективным оказывается объектно-ориентированный подход. Поэтому многие «вещи» оформляются в виде классов.

Пример:

```
from tkinter import *

class But_print:
    def _init_(self):
        self.but = Button(root)
        self.but["text"] = "Печать"
        self.but.bind("<Button-1>",
        self.printer) self.but.pack()
    def printer(self,event):
        print ('Hello World!')

root = Tk()

obj = But_print()

root.mainloop()
```