قــســم الانـظــمــة الـطـبـيـة الـذكــيــة

المرحلة الثانية

Lecture: (6)

Subject: Object oriented programming II
Class: Second
Lecturers: Dr. Dunia H. Hameed , Dr. Maytham N. Meqdad

Object Oriented Programming (II) – Sixth Lecture

Python - GUI Programming (Tkinter)

Python provides various options for developing graphical user interfaces (GUIs). Most important are listed below.

1. Tkinter − Tkinter is the Python interface to the Tk GUI toolkit shipped with Python.
2. wxPython − This is an open-source Python interface for wxWindows http://wxpython.org.
3. JPython − JPython is a Python port for Java which gives Python scripts seamless access to Java class libraries on the local machine http://www.jython.org.

There are many other interfaces available, which you can find them on the net.

Tkinter Programming

Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit. Creating a GUI application using Tkinter is an easy task. All you need to do is perform the following steps −

- Import the Tkinter module.
- Create the GUI application main window.
- Add one or more of the above-mentioned widgets to the GUI application.
- Enter the main event loop to take action against each event triggered by the user.

Example

```
#!/usr/bin/python

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

This would create a following window –

Tkinter Widgets

Tkinter provides various controls, such as buttons, labels and text boxes used in a GUI application. These controls are commonly called widgets.

There are currently 19 types of widgets in Tkinter. We present these widgets as well as a brief description in the following table –

Sr.No.	Operator & Description
1	Button ↗ The Button widget is used to display buttons in your application.
2	Canvas ↗ The Canvas widget is used to draw shapes, such as lines, ovals, polygons and rectangles, in your application.
3	Checkbutton ↗ The Checkbutton widget is used to display a number of options as checkboxes. The user can select multiple options at a time.
4	Entry ↗ The Entry widget is used to display a single-line text field for accepting values from a user.
5	Frame ↗ The Frame widget is used as a container widget to organize other widgets.
6	Label ↗ The Label widget is used to provide a single-line caption for other widgets. It can also contain images.
7	Listbox ↗ The Listbox widget is used to provide a list of options to a user.
8	Menubutton ↗ The Menubutton widget is used to display menus in your application.

9	Menu ⤴
	The Menu widget is used to provide various commands to a user. These commands are contained inside Menubutton.
10	Message ⤴
	The Message widget is used to display multiline text fields for accepting values from a user.
11	Radiobutton ⤴
	The Radiobutton widget is used to display a number of options as radio buttons. The user can select only one option at a time.
12	Scale ⤴
	The Scale widget is used to provide a slider widget.

13	Scrollbar ⤴
	The Scrollbar widget is used to add scrolling capability to various widgets, such as list boxes.
14	Text ⤴
	The Text widget is used to display text in multiple lines.
15	Toplevel ⤴
	The Toplevel widget is used to provide a separate window container.
16	Spinbox ⤴
	The Spinbox widget is a variant of the standard Tkinter Entry widget, which can be used to select from a fixed number of values.
17	PanedWindow ⤴
	A PanedWindow is a container widget that may contain any number of panes, arranged horizontally or vertically.

18	LabelFrame ☑
	A labelframe is a simple container widget. Its primary purpose is to act as a spacer or container for complex window layouts.
19	tkMessageBox ☑
	This module is used to display message boxes in your applications.

1- Button

The Button widget is used to add buttons in a Python application. These buttons can display text or images that convey the purpose of the buttons. You can attach a function or a method to a button which is called automatically when you click the button.

Example:

```python
import Tkinter
import tkMessageBox

top = Tkinter.Tk()

def helloCallBack():
    tkMessageBox.showinfo( "Hello Python", "Hello World")

B = Tkinter.Button(top, text ="Hello", command = helloCallBack)

B.pack()
top.mainloop()
```

2- Canvas

The Canvas is a rectangular area intended for drawing pictures or other complex layouts. You can place graphics, text, widgets or frames on a Canvas.

3- Checkbutton

The Checkbutton widget is used to display a number of options to a user as toggle buttons. The user can then select one or more options by clicking the button corresponding to each option. You can also display images in place of text.

Example:

```python
from Tkinter import *
import tkMessageBox
import Tkinter

top = Tkinter.Tk()
CheckVar1 = IntVar()
CheckVar2 = IntVar()
C1 = Checkbutton(top, text = "Music", variable = CheckVar1, \
                 onvalue = 1, offvalue = 0, height=5, \
                 width = 20)
C2 = Checkbutton(top, text = "Video", variable = CheckVar2, \
                 onvalue = 1, offvalue = 0, height=5, \
                 width = 20)
C1.pack()
C2.pack()
top.mainloop()
```

When the above code is executed, it produces the following result −

4- Entry

The Entry widget is used to accept single-line text strings from a user.

- If you want to display multiple lines of text that can be edited, then you should use the Text widget.

- If you want to display one or more lines of text that cannot be modified by the user, then you should use the Label widget.

Example

```
from Tkinter import *

top = Tk()
L1 = Label(top, text="User Name")
L1.pack( side = LEFT)
E1 = Entry(top, bd =5)
E1.pack(side = RIGHT)

top.mainloop()
```

When the above code is executed, it produces the following result –

5- Frame

The Frame widget is very important for the process of grouping and organizing other widgets in a somehow friendly way. It works like a container, which is responsible for arranging the position of other widgets.
It uses rectangular areas in the screen to organize the layout and to provide padding of these widgets. A frame can also be used as a foundation class to implement complex widgets.