

Tkinter ile Python Arayüz Programlama Ve İncelenmesi

Yunus Emre ÖZTEKİN
Bursa Teknik Üniversitesi, 18360859041@ogrenci.btu.edu.tr

Özetçe- Tkinter, python tarafından yüklü olarak gelen bir arayüz geliştirme kitidir. Tkinter’ı komutları ve Tkinter ile bağlantılı olan modüller üzerine bilgi verilmiştir. Diğer Python arayüz geliştirme kitlerinden olan PyQt5 gibi arayüzlerle karşılaştırılması yapılmıştır. Dezavantajları ve avantajları anlatılan Tkinter için örnek kodlar ve bu kodların çıktıları dokümanda bulunmaktadır. Öğrenilenler doğrultusunda bir demo proje geliştirilmiş olup gerekli bilgileri paylaşılmıştır.

Anahtar Kelimeler- Arayüz, GUI, Widget,

1 GİRİŞ

“Tk” birçok programlama dili tarafından GUI tasarlamak için kullanılan açık kaynaklı, multi-platform bir toolkittir. Tkinter da Python için Tk’ye bir kapıdır, bir arayüzdür. Tkinter tabi nihayetinde bir, Python modülüdür. Fakat kaynak koduna bakıldığında görülür ki, aslında Tcl/Tk kütüphanelerini kullanan bir C uzantısıdır. Küçük masaüstü uygulamalarından, bilimsel modellemelere kadar birçok uygulama alanı vardır. Python ve GUI okumalarındaysanız, Tkinter nispeten işinizi halletmeniz için hızlı bir yol sunabiliyor. Amacımız da zaten tüm bu Python ’da arayüz tasarımı konseptinin incelenmesi olacak.

2.1 TKİNTER WİDGET ’LARI

Tkinter, bir GUI uygulamasında kullanılan düğmeler, etiketler ve metin kutuları gibi çeşitli kontroller sağlar. Bu kontroller genellikle widget olarak adlandırılır.

Button Widget

Düğme pencere ögesidir, uygulamamızdaki düğmeleri görüntülemek için kullanılır.

Canvas Widget

Tuval widget’ı uygulamamızda çizgiler, ovaler, çokgenler ve dikdörtgenler gibi şekiller çizmek için kullanılır.

Checkbox Widget

Onaylama düğmesi widget’ı bir dizi seçeneği onay kutuları olarak görüntülemek için kullanılır. Kullanıcı aynı anda birden fazla seçenek seçebilir.

Entry Widget

Entry yani giriş bilgisi alan pencere ögesi, bir kullanıcıdan değerleri kabul etmek için tek satırlık bir metin alanı görüntülemek için kullanılır.

Label Widget

Label widget’ı diğer widgetlar için tek satırlık bir başlık sağlamak için kullanılır. Türkçe anlamında etikettir. Ayrıca görüntü içerebilir.

Listbox Widget

Liste seçim widget’ı bir kullanıcıya seçenek listesi sağlamak için kullanılır.

Menu Button Widget

Menu düğmesi pencere ögesidir, uygulamamızda menüleri görüntülemek için kullanılır.

Menu Widget

Menü widget’ı bir kullanıcıya çeşitli komutlar sağlamak için kullanılır.

Message Widget

Mesaj widget’ı, bir kullanıcıdan değerleri kabul etmek için çok satırlı metin alanlarını görüntülemek için kullanılır.

Radio Button Widget

Bir dizi seçeneği radyo düğmeleri olarak görüntülemek için kullanılır. Kullanıcı bir seferde yalnızca bir seçim yapabilir.

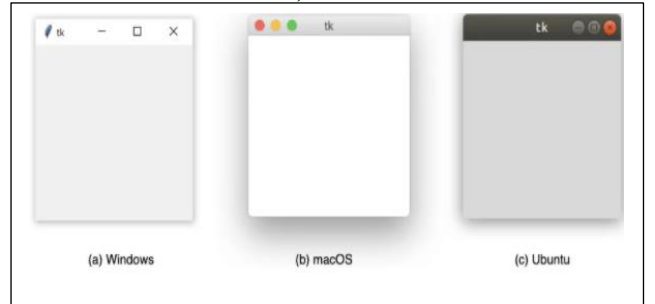
2.2 Örnek Kodlar ve Çıktıları

Bu bölümde Tkinter ile bir önceki bölümde bahsettiğimiz widgetlarımızın mantığını, kodlarını ve çıktılarından örnekler ile anlatımını sağlayacağız.

2.1 İlk Pencere

```
import tkinter as tk
window = tk.Tk()
window.mainloop()
```

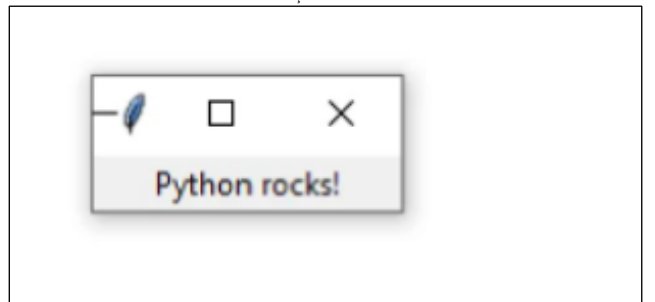
Şekil 1.1



2.2 İlk Label

```
import tkinter as tk
window = tk.Tk()
label = tk.Label(text="Python rocks!")
label.pack()
window.mainloop()
```

Şekil 1.2



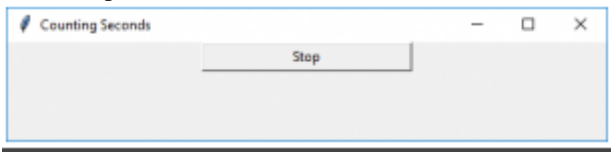
2.3 Düğme Ekleme

```
import tkinter as tk
r = tk.Tk()
```

```

r.title('Counting Seconds')
button = tk.Button(r, text='Stop', width=25,
command=r.destroy)
button.pack()
r.mainloop()

```



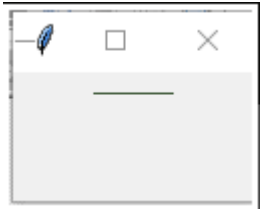
Şekil 1.3

2.4 Canvas Oluşturma

```

from tkinter import *
master = Tk()
w = Canvas(master, width=40, height=60)
w.pack()
canvas_height=20
canvas_width=200
y = int(canvas_height / 2)
w.create_line(0, y, canvas_width, y )
mainloop()

```



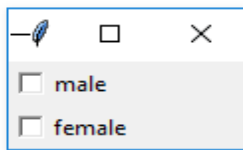
Şekil 1.4

2.5 Check Buton Ekleme

```

from tkinter import *
master = Tk()
var1 = IntVar()
Checkbutton(master, text='male',
variable=var1).grid(row=0, sticky=W)
var2 = IntVar()
Checkbutton(master, text='female',
variable=var2).grid(row=1, sticky=W)
mainloop()

```



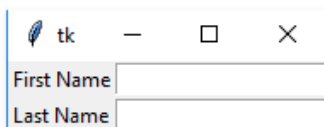
Şekil 1.5

2.6 Entry Oluşturmak

```

from tkinter import *
master = Tk()
Label(master, text='First Name').grid(row=0)
Label(master, text='Last Name').grid(row=1)
e1 = Entry(master)
e2 = Entry(master)
e1.grid(row=0, column=1)
e2.grid(row=1, column=1)
mainloop()

```



Şekil 1.6

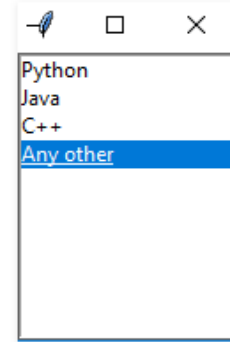
2.7 Listbox Ekleme

```
from tkinter import *
```

```

top = Tk()
Lb = Listbox(top)
Lb.insert(1, 'Python')
Lb.insert(2, 'Java')
Lb.insert(3, 'C++')
Lb.insert(4, 'Any other')
Lb.pack()
top.mainloop()

```



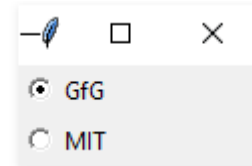
Şekil 1.7

2.8 RadioButton Ekleme

```

from tkinter import *
root = Tk()
v = IntVar()
Radiobutton(root, text='GfG', variable=v,
value=1).pack(anchor=W)
Radiobutton(root, text='MIT', variable=v,
value=2).pack(anchor=W)
mainloop()

```



Şekil 1.8

2.3 Tkinter ve Diğer Python Modülleri

Tkinter bir arayüz modülü olduğu için görsel dökülmesini istediğimiz her şey ile koordineli bir şekilde çalışma imkanı sunuyor. Çok basit ve direkt olarak pythonla yüklü gelmesi bilimsel projelerde ve multi-platform odaklı projelerde kullanımı arttırıyor. Görsel olarak eksileri olsa da gömülü sistemlerden tutun grafik çizdirmeye kadar çokça modül ile beraber kullanımı bulunmaktadır. Bu modüllerden bir kaçını beraber inceleyelim.

2.3.1 Tkinter ve Matplotlib

Matplotlib Python kabuğunda kullanıldığında, grafikler varsayılan bir pencerede görüntülenir. Grafikler wxpython, pygtk veya Tkinter gibi birçok grafik kullanıcı arayüzüne yerleştirilebilir. Çıktı grafiği için hedef olarak kullanılan bu çeşitli seçenekler 'backends' olarak adlandırılır. Backends'i seçmek için Matplotlib.backend içinde çeşitli modüller mevcuttur. Bu modül Tkinter da görüntülenmesi için backend_tkagg'dır.

İlk olarak, Figure() sınıfını kullanarak figure nesnesini

oluşturmamız gerekiyor . Ardından, FigureCanvasTkAgg () sınıfı kullanılarak bir Tkinter tuvali (şekli içeren) oluşturulur . Matplotlib grafikleri varsayılan olarak altta bir araç çubuğuna sahiptir. Ancak Tkinter ile çalışırken , bu araç çubuğunun NavigationToolbar2Tk() sınıfı kullanılarak tuvale ayrı olarak gömülmesi gerekir. O zaman kodumuzu inceleyelim. Şekil 2.1 'de kodumuz görünmektedir.

```
from tkinter import *
from matplotlib.figure import Figure
from matplotlib.backends.backend_tkagg import (FigureCanvasTkAgg, NavigationToolbar2Tk)

def plot():

    fig = Figure(figsize = (5, 5), dpi = 100)

    y = [i**2 for i in range(101)]

    plot1 = fig.add_subplot(111)

    plot1.plot(y)
    canvas = FigureCanvasTkAgg(fig, master = window)
    canvas.draw()

    canvas.get_tk_widget().pack()

    toolbar = NavigationToolbar2Tk(canvas, window)
    toolbar.update()

    canvas.get_tk_widget().pack()

window = Tk()

window.title('Plotting in Tkinter')

window.geometry("500x500")

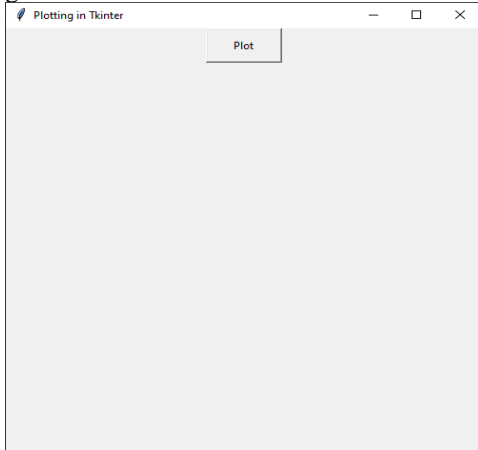
plot_button = Button(master = window, command = plot, height = 2, width = 10, text = "Plot")

plot_button.pack()

window.mainloop()
```

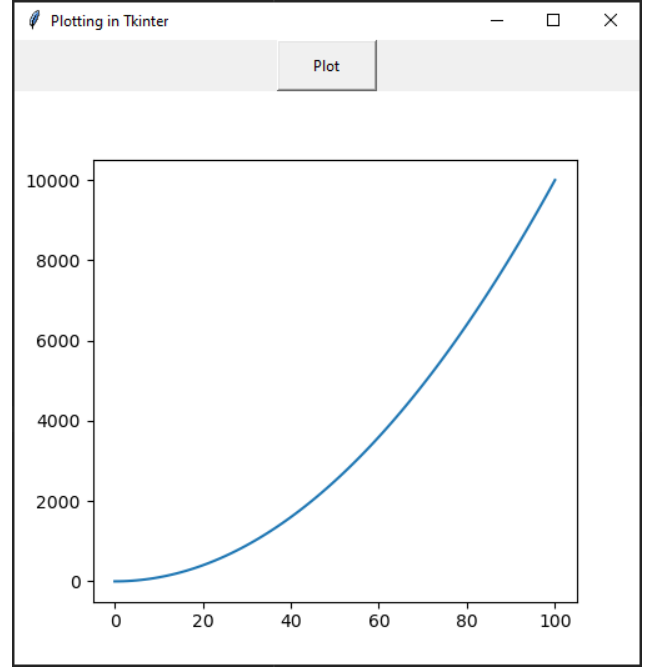
Şekil 2.1

Oluşturduğumuz plot butonuna tıklamadan önce herhangi bir grafik çizimi önümüze düşmemekte. Şekil 2.2 'de gösterilmektedir.



Şekil 2.2

Plot butonuna tıkladıktan oluşturduğumuz fonksiyon çalışıyor ve grafiğimiz görsel arabirimize çizdirilmesini sağlıyor. Şekil 2.3 'te gösterilmektedir.



Şekil 2.3

2.4 Tkinter Vs PyQt5

Hem Tkinter hem de PyQt, kabul edilebilir GUI'leri tasarlamak için kullanışlıdır, ancak aynı zamanda, uyarlanabilirlik ve işlevsellik açısından farklılık gösterirler. Çoğunlukla, Tkinter tamamen GUI yazmak, ayarlarınızı veya işlevselliği aynı komut dosyası içinde programlamakla ilgilidir.

Öte yandan, PyQt'da, bir komut dosyasında GUI'yi ayırırsınız ve Python bilginizi başka bir komut dosyasından kullanırsınız.

Kullanıcı arabirimi için kendi kodunuzu oluşturmak yerine, uygulamanızı geliştirmek için Qt Designer işlevlerini basitçe kullanabilirsiniz.

PyQt kullanmanın avantajları

Kodlama esnekliği - Qt ile GUI programlama, nesneler arasında iletişim kurmak için sinyaller ve yuvalar kavramı etrafında tasarlanmıştır . Bu, GUI olaylarıyla uğraşırken esneklik sağlar ve daha yumuşak bir kod tabanı sağlar.

Bir çerçeveden daha fazlası - Qt, ağ oluşturma, veritabanı oluşturma ve çok daha fazlası için çok çeşitli yerel platform API'leri kullanır . Benzersiz bir API aracılığıyla onlara birincil erişim sunar.

Çeşitli UI bileşenleri - Qt, tümü desteklenen tüm platformlarda temel bir görünümle tasarlanan düğmeler veya menüler gibi çeşitli widget'lar sunar .

Çeşitli öğrenme kaynakları - PyQt, Python için en çok kullanılan UI çerçevelerinden biri olduğu için, çok çeşitli belgelere kolay erişim sağlayabilirsiniz.

PyQt kullanmanın dezavantajları

PyQt5'teki sınıflar için Python'a özgü dokümantasyon eksikliği

PyQt'un tüm ayrıntılarını anlamak için çok zaman gerekir, yani oldukça dik bir öğrenme eğrisi.

Tkinter kullanmanın avantajları

Ticari kullanım için ücretsiz olarak mevcuttur.

Temel Python kitaplığında yer almaktadır.

Tkinter uygulamaları için yürütülebilir dosyalar oluşturmak, Tkinter Python'a dahil edildiğinden daha

erişilebilirdir ve sonuç olarak başka hiçbir bağımlılık içermez.

Tkinter, Python komut dosyaları için hızlı GUI'ler oluşturmak için birincil seçenek olan basit bir API'ye sahip sınırlı bir kitaplık olduğundan anlaşılması ve uzmanlaşması basittir.

Tkinter kullanmanın dezavantajları

Tkinter, gelişmiş gereçler içermez.

Tkinter için Qt Designer ile benzer bir aracı yoktur.

Native bir görünümü ve hissi yok

2.5 Model Uydu Yer İstasyonu Demosu

Model uydu geliştiren bir ekibin yer istasyonu denemeleri doğrultusunda istenilen verileri grafiklerde görünmesi ve anlık grafik çizimi üzerine ihtiyacından doğan bir projedir. Yer istasyonu bu projede demo aşamasında olup canlı verileri anlık alabileceği bir gömülü sistem bulunmadığından Pandas kütüphaneleri ile oluşturulan veri setleri "DataFrame" 'ler aracılığıyla python'da bulunan grafik çizdirme ve gösterme modülüne iletilmektedir. Projemizde hem öğreticilik hem de istenilen verilen çeşitli grafikler altında gösterilebileceğini göstermek için 3 çeşit grafik gösterim şekli kullanılmaktadır. Proje sıcaklık, yükseklik, anlık ve basınç gibi değerleri alacak olsa da biz verilerimizi TÜİK ve benzeri resmî kurumların datalarından alarak oluşturduk.

Oluşturduğumuz datalarımız şekil 3.1, 3.2 ve 3.3 'te görülmektedir.

```
data1 = {'Yillar': ['2002', '2003', '2004', '2005', '2006', '2007', '2008', '2009'],
        'GSYH': [238, 312, 405, 501, 552, 676, 764, 645]}
df1 = DataFrame(data1, columns=['Yillar', 'GSYH'])
```

Şekil 3.1

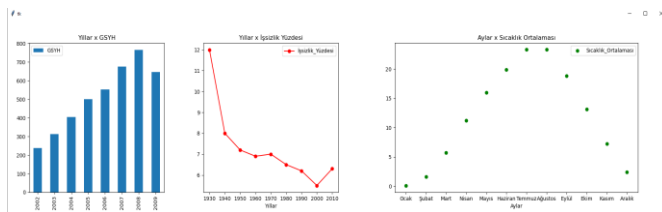
```
data2 = {'Yillar': ['1930', '1940', '1950', '1960', '1970', '1980', '1990', '2000', '2010'],
        'İssizlik_Yuzdesi': [12.8, 8.7, 2.6, 9.7, 6.5, 6.2, 5.5, 6.3]}
df2 = DataFrame(data2, columns=['Yillar', 'İssizlik_Yuzdesi'])
```

Şekil 3.2

```
data3 = {'Ayilar': ['Ocak', 'Şubat', 'Mart', 'Nisan', 'Mayıs', 'Haziran', 'Temmuz', 'Ağustos', 'Eylül', 'Ekim', 'Kasım', 'Aralık'],
        'Sıcaklik_Ortalama': [0.1, 1.6, 5.7, 11.2, 16.0, 19.9, 23.3, 23.3, 18.0, 13.1, 7.2, 2.4]}
df3 = DataFrame(data3, columns=['Ayilar', 'Sıcaklik_Ortalama'])
```

Şekil 3.3

Programımızı çalıştırdığımız zaman grafiklerimizin istediğimiz şekilde çizildiğini hepsini tek bir tk penceresi altında görebildiğimizi görüyoruz. (Şekil 3.4)



Şekil 3.4

2.6 SONUÇ

Şekil 3.4 üzerinde de görüldüğü üzere Tkinter ile diğer modüllerin etkileşimini ve Tkinter'ın kullanım kolaylığı

hakkında bilgi verdik ve bununla bir proje Demosu dahi oluşturduk. Yaptığımız çalışmalar sonucunda Tkinter'ın diğer arayüz geliştirme modülleri arasındaki farklara değindik, kimilerinin avantajlarından kimilerinin dezavantajından bahsettik. Özel bir yargı olmakla beraber Tkinter arayüz geliştirme kitinin giriş seviyesi için çok ideal bir araç olduğunu elimizdeki verilerle de sununldu.

3. KAYNAKLAR

Mckinney, W (2015). Pandas, Python Data Analysis Library. Available at <http://pandas.pydata.org/>.

Hunter, J D (2007). Matplotlib: A 2D Graphics Environment. *Comput Sci Eng* May 20079(3): 90-95, DOI: <https://doi.org/10.1109/MCSE.2007.55>

Astronomical Data Analysis Software and Systems XIV ASP Conference Series, Vol. 347, Proceedings of the Conference held 24-27 October, 2004 in Pasadena, California, USA. Edited by P. Shopbell, M. Britton, and R. Ebert. San Francisco: Astronomical Society of the Pacific, 2005., p.91

JOHN E. GRAYSON(2010) Python and Tkinter Programming, 29-315