ALGORİTMALAR VE PROGRAMLAMA

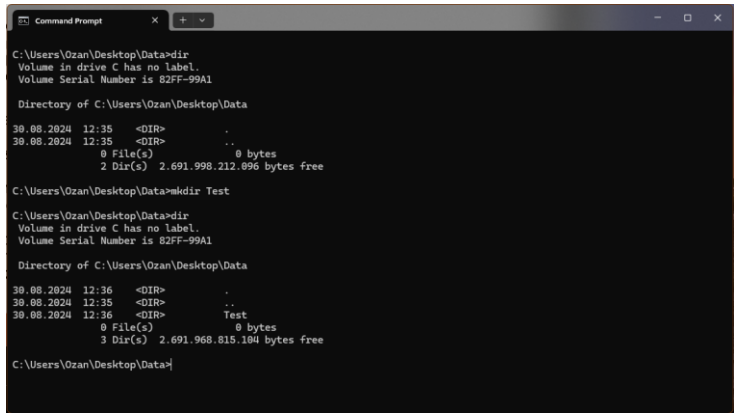
Tkinter MODÜL UYGULAMALARI

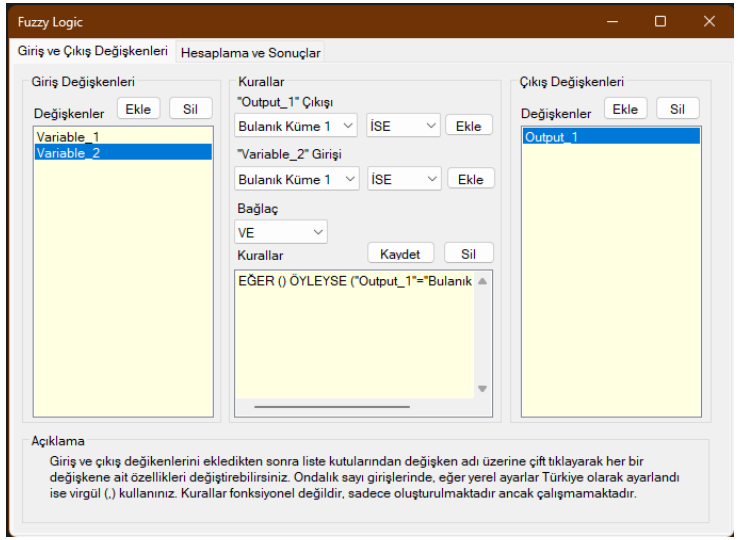
Dr. Öğr. Üyesi M. Ozan AKI

Tkinter MODÜL UYGULAMALARI

Şimdiye kadar yazdığımız programlar terminal ekranında (konsol) çalışmaktaydı. Bu ara yüze ayrıca Komut İstemi Arayüzü (**CLI**: Command Line Interface) de denir.

Ancak son kullanıcıya yönelik yazılımlarda genelde görsel arayüz kullanılır. İngilizcesi Graphical User Interface (**GUI**) olan Grafiksel Kullanıcı Arayüzü, kullanıcıların komut satırı kodlarını ezberlemeden, fare ve dokunmatik ekran gibi girdi araçları sayesinde pencereler içinde yer alan butonlar, metin kutuları, liste kutuları gibi nesneleri kullanarak bilgisayarları kontrol etmelerini sağlar.





Tkinter MODÜL UYGULAMALARI

Şimdiye kadar yazdığımız programlar terminal ekranında (konsol) çalışmaktaydı. Bu ara yüze ayrıca Komut İstemi Arayüzü (**CLI**: Command Line Interface) de denir.

Ancak son kullanıcıya yönelik yazılımlarda genelde görsel arayüz kullanılır. İngilizcesi Graphical User Interface (**GUI**) olan Grafiksel Kullanıcı Arayüzü, kullanıcıların komut satırı kodlarını ezberlemeden, fare ve dokunmatik ekran gibi girdi araçları sayesinde pencereler içinde yer alan butonlar, metin kutuları, liste kutuları gibi nesneleri kullanarak bilgisayarları kontrol etmelerini sağlar.

GUI sistemlerin CLI sistemlerden temel farkı, olay güdümlü (**event-driven**) çalışmalarıdır. Örneğin CLI sistemde vize, final notları belirli bir sıra ile kullanıcıdan istenip, sonrasında hesaplama işlemi yapılırken, GUI sistemde ise kullanıcı ilgili metin kutularına istediği sırada bu verileri girip sonra istediği zaman bir butona tıklayarak hesaplatır.
Bu örnekteki butonun tıklanması bir olaydır (**event**) ve her olay belirli bir kod bloğu ile ilişkilendirilir.

Tkinter MODÜL UYGULAMALARI

tkinter modülü, görsel arayüz oluşturmak için metotlar sağlayan bir modüldür.

tkinter, Python ile birlikte yüklenir ayrıca bir kurulum gerektirmez.

GUI sisteminde kullanılan buton, metin kutuları gibi kontrollere **widget** adı verilir.

Bazı Widget ler

Button
Checkbutton
Frame
LabelFrame
Menu
Message
PanedWindow
Scale
Spinbox
tkMessageBox
Canvas
Entry
Label
ListBox
Menubutton
Optionmenu
Radiobutton
Scrollbar
Text

Tkinter MODÜL UYGULAMALARI

tkinter modülü **import** edildikten sonra bir pencere (**window**) nesnesi

değişken=**Tk()**

Metodu ile oluşturulur.

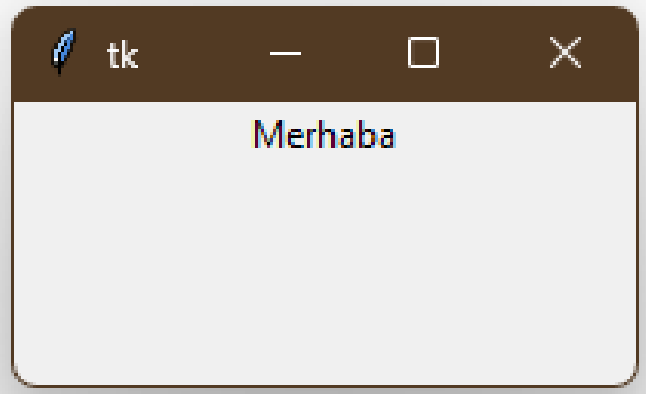
Elde edilen değişken, pencere içerisine yerleştirilmek istenen widget metotlarında parametre olarak kullanılır.

Örneğin, “Merhaba” yazısını bir **Label widget** olarak pencereye eklemek için;

yazi=**Label(***değişken*,**text=**“Merhaba”**)**

yazi.**pack()**

```
>>> import tkinter as tk
>>>
>>> pencere=tk.Tk()
>>>
>>> yazi=tk.Label(pencere, text="Merhaba")
>>> yazi.pack()
>>>
```



Tkinter MODÜL UYGULAMALARI

Pencereye bir **Buton widget** eklemek için;

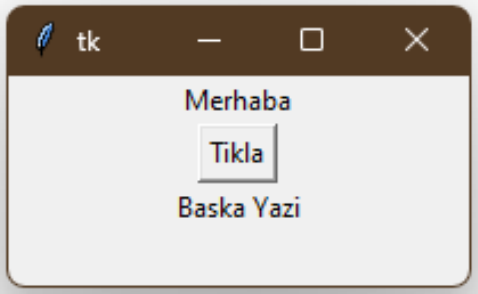
buton=**Buton**(*pencere*,**text=**"Tikla")

buton.**pack()**

Benzer şekilde, başka kontroller eklemek için benzer şekilde her bir **widget** için, pencere değişkeninin parametre olarak verildiği bir nesne oluşturulur.

Eklenen her **widget**, pencerenin üstünden aşağıya doğru sırasıyla bitişik olarak oluşturulur.

```
>>> from tkinter import *
>>>
>>> pencere=Tk()
>>>
>>> yazi=Label(pencere, text="Merhaba")
>>> yazi.pack()
>>>
>>> import tkinter as tk
>>>
>>> pencere=tk.Tk()
>>>
>>> yazi=tk.Label(pencere, text="Merhaba")
>>> yazi.pack()
>>>
>>> buton=tk.Button(pencere, text="Tikla")
>>> buton.pack()
>>>
>>> yazi2=tk.Label(pencere, text="Baska Yazi")
>>> yazi2.pack()
>>>
```



Tkinter MODÜL UYGULAMALARI

Pencereye eklenen **widget** konuları değiştirilmek istenirse, **pack** metoduna **side** parametresi ile yer bilgisi verilebilir.;

buton1=**Buton(***pencere*,**text=**"Solda")

buton1.**pack(side=LEFT)**

buton2=**Buton(***pencere*,**text=**"Sagda")

buton2.**pack(side=RIGHT)**

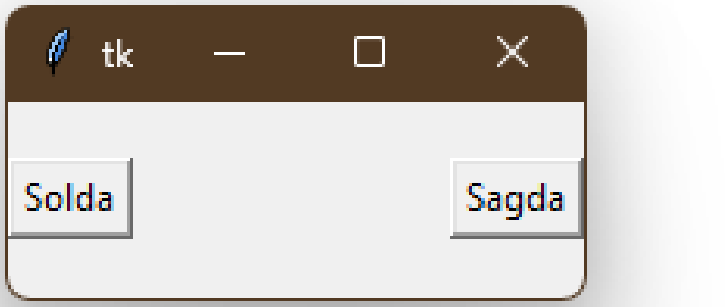
Benzer şekilde;

Side=TOP **side=LEFT**

Side=BOTTOM **side=RIGT**

İfadeleri kullanılabilir.

```
>>> import tkinter as tk
>>>
>>> pencere=tk.Tk()
>>> buton1=tk.Button(pencere,text="Solda")
>>> buton1.pack(side=tk.LEFT)
>>> buton2=tk.Button(pencere,text="Sagda")
>>> buton2.pack(side=tk.RIGHT)
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
```



Tkinter MODÜL UYGULAMALARI

Eğer konrolleri pencereye belirli bir konuma yerleştirmek isterse;

buton1=**Buton(***pencere*,**text=**"Solda"**)**

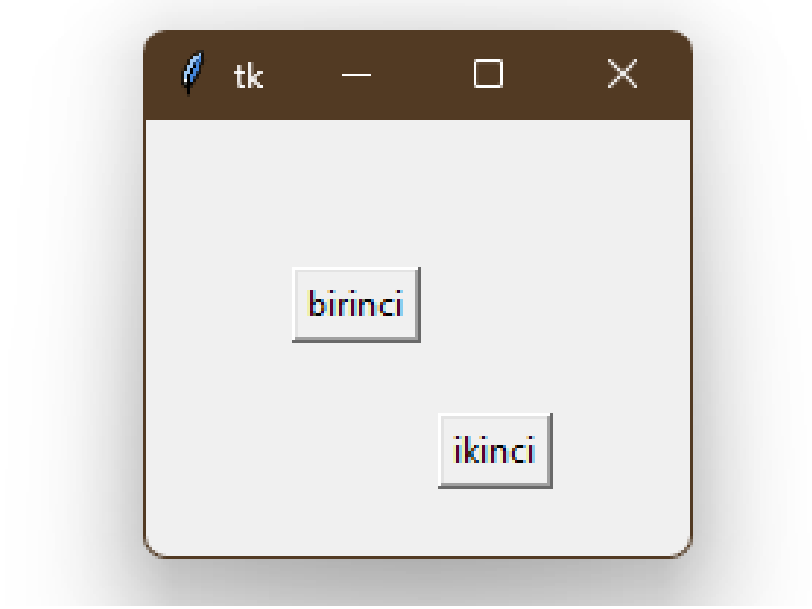
buton1.**place(x=100, y=100)**

Şeklinde x ve y koordinatlarını girerek widget konumu belirtilir.

Koordinatın 0,0 noktası pencereni üst sol köşesidir. X yatayda sağa doğru arttarken y dikeyde aşağı doğru artar.

Pencere boyutlarını aşan koordinatlar pencerede görünmez.

```
>>> import tkinter as tk
>>>
>>> pencere=tk.Tk()
>>> buton1=tk.Button(pencere,text="birinci")
>>> buton1.place(x=50,y=50)
>>> buton2=tk.Button(pencere,text="ikinci")
>>> buton2.place(x=100,y=100)
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
```

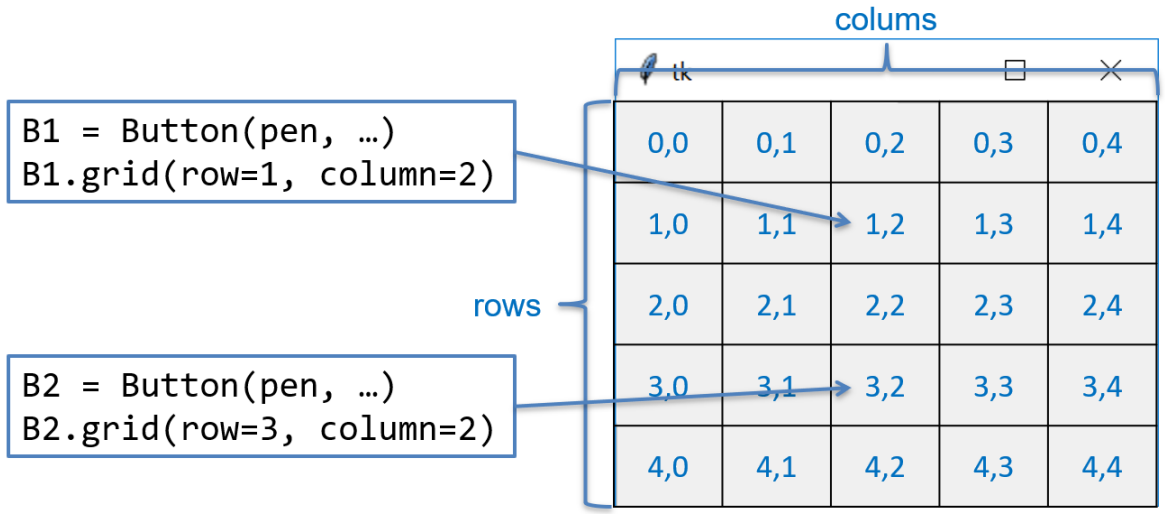


Tkinter MODÜL UYGULAMALARI

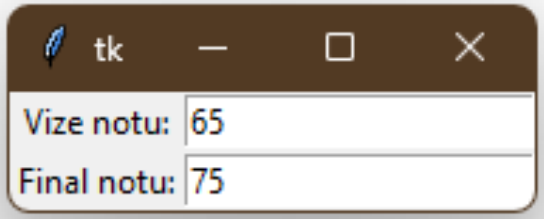
Pencereyi sanal olarak satır ve sütunlara bölerek istediğimiz hücrelere satır ve sütun bilgisi girerek **widget** yerleştirmek için;

grid(row=satır**, column=**sütun**)**

Metodu kullanılır. Böylece yatay ve dikey olarak hizalanmış bir görsel çıktı elde edebiliriz.




```
>>> import tkinter as tk
>>>
>>> pencere=tk.Tk()
>>> label1=tk.Label(pencere,text="Vize notu:")
>>> label1.grid(row=0,column=0)
>>> label2=tk.Label(pencere,text="Final notu:")
>>> label2.grid(row=1,column=0)
>>> entry1=tk.Entry(pencere)
>>> entry1.grid(row=0,column=1)
>>> entry2=tk.Entry(pencere)
>>> entry2.grid(row=1,column=1)
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
```



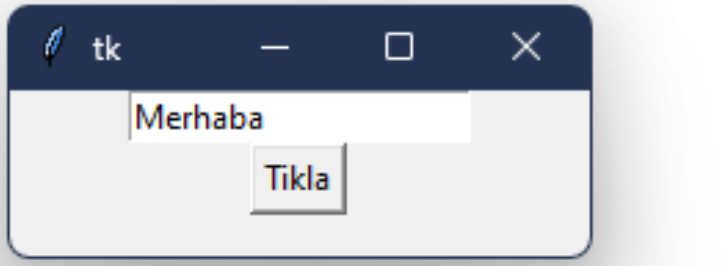
Tkinter MODÜL UYGULAMALARI

Bir butona tıklandığında, daha önce hazırlamış olduğumuz fonksiyonun çağırılmasını sağlamak için, **Button widget** parametresi olarak;

command=fonksiyon_adı

Parametresini eklememiz gerekir. Böylece butona tıklandığında, ilişkilendirdiğimiz kodlar otomatik olarak çalıştırılacaktır.

```
>>> import tkinter as tk
>>>
>>> def yazi_ekle():
...     entry.insert(0,"Merhaba")
...
>>> pencere=tk.Tk()
>>> entry=tk.Entry(pencere)
>>> entry.pack()
>>> buton=tk.Button(pencere,text="Tikla",command=yazi_ekle)
>>> buton.pack()
```



Tkinter MODÜL UYGULAMALARI

Buraya kadar gösterilen **widget** bileşenleri ile basit bir uygulama yazabiliriz.

Vize, Final ve Ortalama kutucukları ile bir Hesapla butonu olan pencerede, ilgili kutucuklara vize ve final notları girildiğinde ve ardından hesapla butonuna tıklandığında, ortalamayı hesaplayarak ortalama kutucuğunda gösteren görsel pencereyi ve yazılımı hazırlayalım.

```
>>> import tkinter as tk
>>>
>>> def hesapla():
...     v=int(vize.get())
...     f=int(final.get())
...     ort=v*0.3+f*0.7
...     ortalama.delete(0,tk.END)
...     ortalama.insert(0,str(ort))
...
>>> pencere=tk.Tk()
>>> label_vize=tk.Label(pencere,text="Vize notu:")
>>> label_vize.grid(row=0,column=0)
>>> label_final=tk.Label(pencere,text="Final notu:")
>>> label_final.grid(row=1,column=0)
>>> label_ort=tk.Label(pencere,text="Ortalama:")
>>> label_ort.grid(row=3,column=0)
>>> vize=tk.Entry(pencere)
>>> vize.grid(row=0,column=1)
>>> final=tk.Entry(pencere)
>>> final.grid(row=1,column=1)
>>> ortalama=tk.Entry(pencere)
>>> ortalama.grid(row=3,column=1)
>>> buton=tk.Button(pencere,text="Hesapla",command=hesapla)
>>> buton.grid(row=2,column=1)
```