

7. Többablakos alkalmazás (Toplevel)

Többablakos alkalmazás akár Tk, akár Toplevel segítségével is készíthető.

Feladat – Több ablak

Készítsünk olyan ablakot, amelynek egy nyomógombjára kattintva egy másik ablak nyílik meg!



```
# -*- coding: ISO-8859-2 -*-  
from Tkinter import *
```

```
abl1 = Tk()
```

```
def ujablak():
```

```
    abl2 = Toplevel(abl1)  
    uz2 = Message(abl2, text='Készítette: Gipsz Jakab\nPiripócs\n2009.06.04', width=300)  
    gomb2 = Button(abl2, text='Kilép', command=abl2.destroy)  
    uz2.pack()  
    gomb2.pack()  
    abl2.mainloop()
```

```
# a widgetek létrehozása
```

```
szoveg1 = Label(abl1, text='Kattints a gombra!')  
gomb1 = Button(abl1, text='Névjegy', command=ujablak)
```

```
# lap tördelés a 'pack' metódus segítségével :
```

```
szoveg1.pack()  
gomb1.pack()
```

```
# indítás :
```

```
abl1.mainloop()
```

Feladat – Több ablak2

Készítsünk olyan alkalmazást, amelyben bekérjük egy téglatest egy csúcsba futó éleit, majd a felszínt és a térfogatot egy új ablakban írjuk ki!



```
# -*- coding: ISO-8859-2 -*-
from Tkinter import *

abl1 = Tk()
abl1.title('A téglatest adatai')
abl1.minsize(width=300, height=100)

def ujablak():
    abl2 = Toplevel(abl1)
    abl2.title('Eredmények')
    abl2.minsize(width=300, height=100)

    # a widgetek létrehozása
    sz1 = Label(abl2, text='Felszín:')
    sz2 = Label(abl2, text='Térfogat:')
    m1 = Entry(abl2)
    m2 = Entry(abl2)

    # lap tördelés a 'grid' módszer segítségével :
    sz1.grid(row=1)
    sz2.grid(row=2)
    m1.grid(row=1, column=2, sticky=W)
    m2.grid(row=2, column=2, sticky=W)

    a = eval(mezo1.get())
    b = eval(mezo2.get())
    c = eval(mezo3.get())

    felszin = 2*(a*b+a*c+b*c)
    terfogat = a*b*c

    m1.delete(0,END)
    m1.insert(0,str(felszin))
    m2.delete(0,END)
    m2.insert(0,str(terfogat))
```

```
abl2.mainloop()
```

```
# a widgetek létrehozása
```

```
szoveg1 = Label(abl1, text='a:')
```

```
szoveg2 = Label(abl1, text='b:')
```

```
szoveg3 = Label(abl1, text='c:')
```

```
gomb1 = Button(abl1, text='Számítás', command=ujablak)
```

```
mezo1 = Entry(abl1)
```

```
mezo2 = Entry(abl1)
```

```
mezo3 = Entry(abl1)
```

```
# lap tördelés a 'grid' módszer segítségével :
```

```
szoveg1.grid(row=1)
```

```
szoveg2.grid(row=2)
```

```
szoveg3.grid(row=3)
```

```
gomb1.grid(row=4, column=2, sticky=W)
```

```
mezo1.grid(row=1, column=2, sticky=W)
```

```
mezo2.grid(row=2, column=2, sticky=W)
```

```
mezo3.grid(row=3, column=2, sticky=W)
```

```
# indítás :
```

```
abl1.mainloop()
```

8. Menükészítés

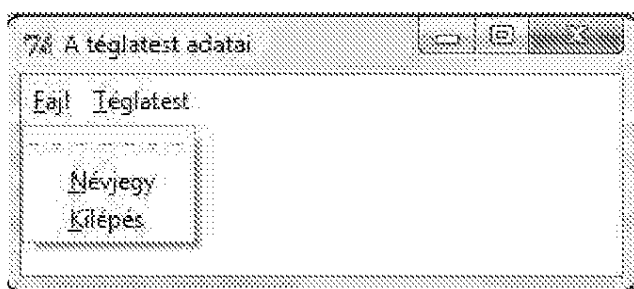
A menük definiálása egy ablakban, egy dedikált *Frame* segítségével történik. A *Menubutton* widget-tel lehet definiálni a menüsáv egy elemét és a *Menu* widget-tel lehet definiálni az elemhez kapcsolt parancsokat.

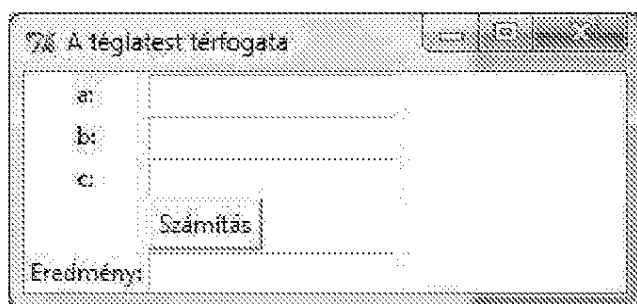
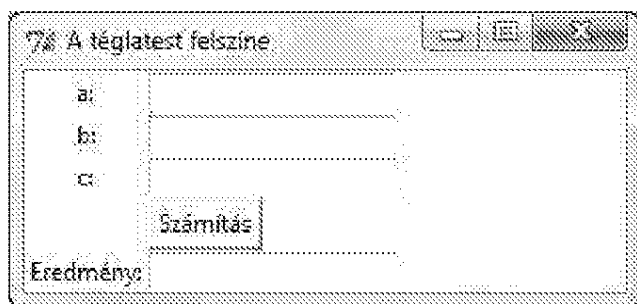
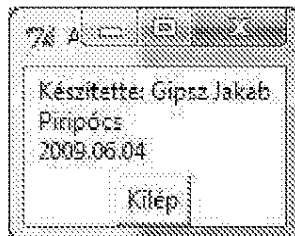
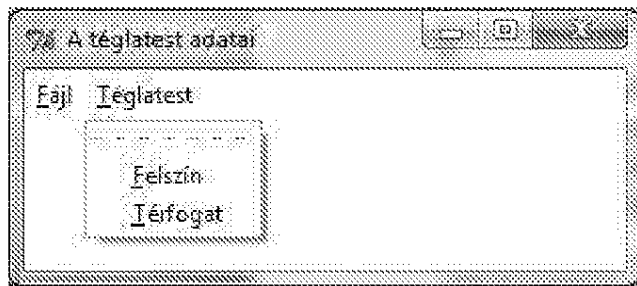
A sávban a menük szervezése a következő változók és függvények segítségével történik:

- **bd, refile** a menüpontok száma és a domborzat (RAISED, SUNKEN, RIDGE),
- **add_separator()** egy elválasztó vonalat illeszt be a menü két parancsa közé,
- **add_cascade()** almenü létrehozása,
- **entryconfig(num, state=)** beállítja az elérhető menüpontok (alapértelmezetten ENABLED), illetve a nem elérhető menüpontok (DISABLED) megjelenését.

Feladat – Menü

Készítsünk olyan alkalmazást, amelyben menük segítenek annak a kiválasztásában, hogy egy téglatest egy csúcsba futó éleiből, annak felszínét vagy a térfogatát számítsuk ki, illetve, hogy a program készítőjének névjegyét tekintsük meg, vagy lépünk ki a programból!





```
# -*- coding: ISO-8859-2 -*-
from Tkinter import *
```

```
#Névjegy ablak
```

```
def nevjegy():
```

```
    abl2 = Toplevel(foablak)
```

```
    uz2 = Message(abl2, text='Készítette: Gipsz Jakab\nPiripócs\n2009.06.04', width=300)
```

```
    gomb2 = Button(abl2, text='Kilép', command=abl2.destroy)
```

```
    uz2.pack()
```

```
    gomb2.pack()
```

```
    abl2.mainloop()
```

```
#Névjegy ablak vége
```

```
#Felszín ablak
```

```
def felszin():
```

def szamit():

a = eval(mezo1.get())

b = eval(mezo2.get())

c = eval(mezo3.get())

felszin = 2(a*b+a*c+b*c)*

mezo4.delete(0,END)

mezo4.insert(0,str(felszin))

abl3 = Toplevel(foablak)

abl3.title('A téglatest felszíne')

abl3.minsize(width=300, height=100)

szoveg1 = Label(abl3, text='a:')

szoveg2 = Label(abl3, text='b:')

szoveg3 = Label(abl3, text='c:')

szoveg4 = Label(abl3, text='Eredmény:')

gomb1 = Button(abl3, text='Számítás', command=szamit)

mezo1 = Entry(abl3)

mezo2 = Entry(abl3)

mezo3 = Entry(abl3)

mezo4 = Entry(abl3)

szoveg1.grid(row=1)

szoveg2.grid(row=2)

szoveg3.grid(row=3)

szoveg4.grid(row=5)

gomb1.grid(row=4, column=2, sticky=W)

mezo1.grid(row=1, column=2, sticky=W)

mezo2.grid(row=2, column=2, sticky=W)

mezo3.grid(row=3, column=2, sticky=W)

mezo4.grid(row=5, column=2, sticky=W)

abl3.mainloop()

#Felszín ablak vége

#Térfogat ablak

def terfogat():

def szamit():

a = eval(mezo1.get())

b = eval(mezo2.get())

c = eval(mezo3.get())

*terfogat = a*b*c*

mezo4.delete(0,END)

mezo4.insert(0,str(terfogat))

abl3 = Toplevel(foablak)

abl3.title('A téglatest térfogata')

abl3.minsize(width=300, height=100)

szoveg1 = Label(abl3, text='a:')

szoveg2 = Label(abl3, text='b:')

szoveg3 = Label(abl3, text='c:')

szoveg4 = Label(abl3, text='Eredmény:')

```

gomb1 = Button(abl3, text='Számítás', command=szamit)
mez1 = Entry(abl3)
mez2 = Entry(abl3)
mez3 = Entry(abl3)
mez4 = Entry(abl3)
szoveg1.grid(row=1)
szoveg2.grid(row=2)
szoveg3.grid(row=3)
szoveg4.grid(row=5)
gomb1.grid(row=4, column=2, sticky=W)
mez1.grid(row=1, column=2, sticky=W)
mez2.grid(row=2, column=2, sticky=W)
mez3.grid(row=3, column=2, sticky=W)
mez4.grid(row=5, column=2, sticky=W)
abl3.mainloop()
#Térfogat ablak vége

#Főablak
foablak = Tk()
foablak.title('A téglatest adatai')
foablak.minsize(width=300, height=100)

menusor = Frame(foablak)
menusor.pack(side=TOP, fill=X)

menu1 = Menubutton(menusor, text='Fájl', underline=0)
menu1.pack(side=LEFT)
fajl = Menu(menu1)
fajl.add_command(label='Névjegy', command=nevjegy, underline=0)
fajl.add_command(label='Kilépés', command=foablak.destroy, underline=0)
menu1.config(menu=fajl)

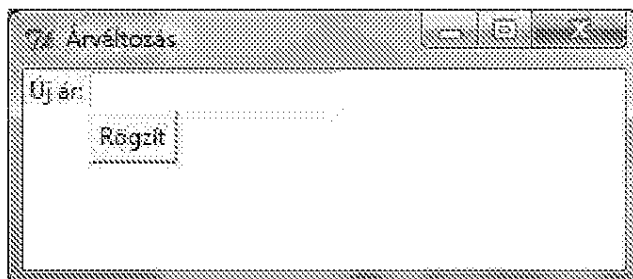
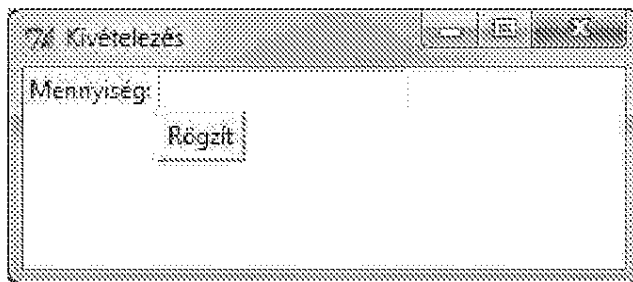
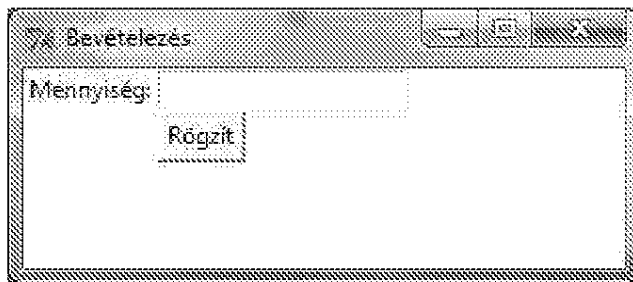
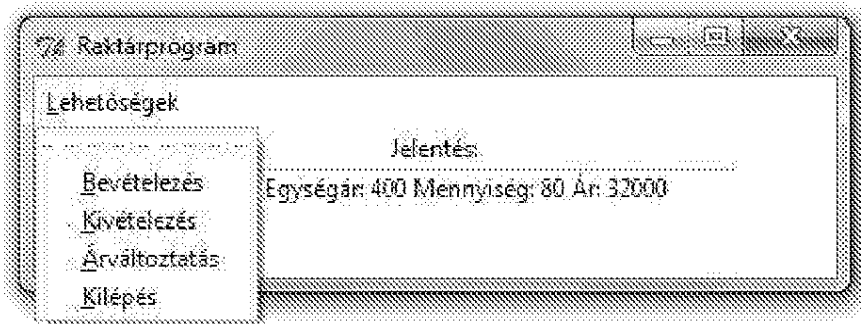
menu2 = Menubutton(menusor, text='Téglatest', underline=0)
menu2.pack(side=LEFT)
teglatest = Menu(menu2)
teglatest.add_command(label='Felszín', command=felszin, underline=0)
teglatest.add_command(label='Térfogat', command=terfogat, underline=0)
menu2.config(menu=teglatest)

foablak.mainloop()

```

Feladat – Raktárprogram

Adott egy zöldségraktár, melyben pillanatnyilag egyetlen árut, paradicsomot raktározunk. A raktárba gyakran teszünk be, illetve veszünk ki onnan paradicsomot. A paradicsom pillanatnyi egységára 300 Ft, de ez változhat. Készítsünk olyan programot, amely segítségével rögzíteni tudjuk a megfelelő adatokat, és bármikor jelentést tudunk adni a paradicsom aktuális mennyiségéről, egységáráról, értékéről! Menüből lehessen kiválasztani, hogy beteszünk-e, vagy kivesszünk paradicsomot, illetve, hogy emeljük, avagy csökkentjük a paradicsom egységárát!



```
# -*- coding: ISO-8859-2 -*-  
from Tkinter import *
```

```
class Aru:  
    def __init__(self, aruNev, aruEgysegar):  
        self.nev = aruNev  
        self.egysegar = aruEgysegar  
        self.menny = 0  
  
    def setEgysegar(self, aruEgysegar):  
        if (aruEgysegar >= 0):  
            self.egysegar = aruEgysegar  
  
    def getAr(self):  
        return self.menny * self.egysegar
```

```

def hozzatesz(self, aruMenny):
    if (aruMenny > 0):
        self.menny = self.menny + aruMenny

def elvesz(self, aruMenny):
    if (aruMenny > 0) and (aruMenny <= self.menny):
        self.menny = self.menny - aruMenny

def __doc__(self):
    mezol.delete(0, END)
    mezol.insert(0, self.nev+' Egységár: '+str(self.egysegar)+' Mennyiség: '+str(self.menny)+'
    Ár: '+str(self.getAr()))

aru = Aru("Paradicsom", 300)

#Bevételezés ablak

def bevetelezes():

    def rogzit():
        menny = eval(m1.get())
        aru.hozzatesz(menny)
        aru.__doc__()
        abl2.destroy()

    abl2 = Toplevel(foablak)
    abl2.title('Bevételezés')
    abl2.minsize(width=300, height=100)
    sz1 = Label(abl2, text='Mennyiség:')
    g1 = Button(abl2, text='Rögzít', command=rogzit)
    m1 = Entry(abl2)
    sz1.grid(row=1)
    g1.grid(row=2, column=2, sticky=W)
    m1.grid(row=1, column=2, sticky=W)
    abl2.mainloop()

#Bevételezés ablak vége

#Kivételezés ablak

def kivetelezes():

    def rogzit():
        menny = eval(m1.get())
        aru.elvesz(menny)
        aru.__doc__()
        abl3.destroy()

    abl3 = Toplevel(foablak)

```



```

abl3.title('Kivételezés')
abl3.minsize(width=300, height=100)
sz1 = Label(abl3, text='Mennyiség:')
g1 = Button(abl3, text='Rögzít', command=rogzit)
m1 = Entry(abl3)
sz1.grid(row=1)
g1.grid(row=2, column=2, sticky=W)
m1.grid(row=1, column=2, sticky=W)
abl3.mainloop()

#Kivételezés ablak vége

#Árváltozás ablak

def arvaltozas():

    def rogzit():
        ujar = eval(m1.get())
        aru.setEgysegar(ujar)
        aru.__doc__()
        abl4.destroy()

    abl4 = Toplevel(foablak)
    abl4.title('Árváltozás')
    abl4.minsize(width=300, height=100)
    sz1 = Label(abl4, text='Új ár:')
    g1 = Button(abl4, text='Rögzít', command=rogzit)
    m1 = Entry(abl4)
    sz1.grid(row=1)
    g1.grid(row=2, column=2, sticky=W)
    m1.grid(row=1, column=2, sticky=W)
    abl4.mainloop()

#Árváltozás ablak vége

#Főablak

foablak = Tk()
foablak.title('Raktárprogram')
foablak.minsize(width=400, height=100)

menusor = Frame(foablak)
menusor.pack(side=TOP, fill=X)

menu1 = Menubutton(menusor, text='Lehetőségek', underline=0)
menu1.pack(side=LEFT)
fajl = Menu(menu1)
fajl.add_command(label='Bevételezés', command=bevetelezes, underline=0)
fajl.add_command(label='Kivételezés', command=kivetelezes, underline=0)
fajl.add_command(label='Árváltoztatás', command=arvaltozas, underline=0)

```

```
fajl.add_command(label='Kilépés', command = foablak.destroy, underline =0)
menu1.config(menu = fajl)
```

```
szoveg1 = Label(foablak, text ='Jelentés:')
mezol = Entry(foablak, width =50)
szoveg1.pack()
mezol.pack()
aru.__doc__()
```

```
foablak.mainloop()
```

Feladat – Síkidomok öröklődése

Írjunk olyan programot, amely kiszámítja a négyzet, a téglalap és a kör kerületét, területét! Az egyes síkidomok osztályok legyenek, megfelelő metódusokkal, alkalmazzuk az öröklődést!

