



1 LYGIS

12 paskaita. Grafinės sąsajos (GUI) kūrimas



Šiandien išmoksite

01

Susipažinsime su Tkinter biblioteka

02

Kurti paprastus GUI objektus

03

Kurti norimus GUI layout'us



Grafinės sąsajos objektai

- Label – užrašas (kortelė)
- Button – mygtukas
- Entry – laukelis
- Menu – meniu
- Frame – rėmelis
- Checkbutton – varnelė
- Listbox – sąrašas
- Scrollbar – sąrašo slinkimo juosta



```
from tkinter import *  
  
langas = Tk()  
langas.geometry("250x200")  
uzrasas = Label(langas, text="Tiesiog tekstas")  
uzrasas.pack()  
langas.mainloop()
```

Minimali programa



```
from tkinter import *

langas = Tk()

virsutinis = Frame(langas)
apatinis = Frame(langas)

mygtukas1 = Button(virsutinis, text="1 mygtukas")
mygtukas2 = Button(virsutinis, text="2 mygtukas")
mygtukas3 = Button(virsutinis, text="3 mygtukas")
mygtukas4 = Button(apatinis, text="4 mygtukas")

virsutinis.pack()
apatinis.pack(side=BOTTOM)
mygtukas1.pack(side=LEFT)
mygtukas2.pack(side=LEFT)
mygtukas3.pack(side=LEFT)
mygtukas4.pack(side=BOTTOM, fill=Y)

langas.mainloop()
```

Grafinių objektų formavimas rėmeliuose (su pack funkcija)



```
from tkinter import *

langas = Tk()

uzrasas1 = Label(langas, text="Vardas")
laukas1 = Entry(langas)
uzrasas2 = Label(langas, text="Pavardė")
laukas2 = Entry(langas)
varnele = Checkbutton(langas, text="Pažymėk varnelę")

uzrasas1.grid(row=0, column=0, sticky=E)
laukas1.grid(row=0, column=1)
uzrasas2.grid(row=1, column=0, sticky=E)
laukas2.grid(row=1, column=1)
varnele.grid(row=2, columnspan=2)

langas.mainloop()
```

Grafinių objektų formavimas lentelėje (su grid funkcija)



```
from tkinter import *
langas = Tk()

def spausdinti():
    print("Spausdina!")

mygtukas = Button(langas, text="Spausdinti", command=spausdinti)
mygtukas.pack()
langas.mainloop()

# Spausdina!
```

Kaip įdėti funkciją paleidžiantį mygtuką



```
from tkinter import *
langas = Tk()

def spausdinti(event):
    print("Paspaustas kairys pelės mygtukas!")

def spausdinti2(event):
    print("Paspaustas dešinys pelės mygtukas!")

def spausdinti3(event):
    print("Paspaustas ENTER!")

mygtukas = Button(langas, text="Spausdinti")
mygtukas.bind("<Button-1>", spausdinti)
mygtukas.bind("<Button-3>", spausdinti2)
langas.bind("<Return>", spausdinti3)
mygtukas.pack()

langas.mainloop()

# Paspaustas kairys pelės mygtukas!
# Paspaustas dešinys pelės mygtukas!
# Paspaustas ENTER!
```

**Kaip esant skirtingiems
vartotojo veiksmams,
paleisti skirtingas funkcijas**



```
from tkinter import *
langas = Tk()

def spausdinti():
    print("Spausdina!")

mygtukas = Button(langas, text="Spausdinti", command=spausdinti)
langas.bind("<Return>", lambda event: spausdinti())
mygtukas.pack()

langas.mainloop()
```

**Kaip per bind iškviesti
funkciją be "event" argumento**



```
from tkinter import *

langas = Tk()

def spausdinti():
    iversta = laukas1.get()
    rezultatas["text"] = iversta

uzrastas1 = Label(langas, text="Įrašykite žodį")
laukas1 = Entry(langas)
mygtukas = Button(langas, text="Įvesti", command=spausdinti)
rezultatas = Label(langas, text="")

uzrastas1.grid(row=0, column=0)
laukas1.grid(row=0, column=1)
mygtukas.grid(row=0, column=2)
rezultatas.grid(row=1, columnspan=3)

langas.mainloop()
```

Kaip per grafinę sąsają nuskaityti ir atspausdinti duomenis



```
from tkinter import *  
  
langas = Tk()  
boksas = Listbox(langas)  
sarajas = range(1, 200)  
boksas.insert(END, *sarajas)  
boksas.pack(side=LEFT)  
langas.mainloop()
```

Kaip sukurti atvaizduojamą sąrašą



```
from tkinter import *
langas = Tk()
masyvas = range(1, 200)
scrollbaras = Scrollbar(langas)
boksas = Listbox(langas,
yscrollcommand=scrollbaras.set)
scrollbaras.config(command=boksas.yview)
boksas.insert(END, *masyvas)
scrollbaras.pack(side=RIGHT, fill=Y)
boksas.pack(side=LEFT)
langas.mainloop()
```

Kaip pridėti sąrašo slinkimo juostą



```
from tkinter import *

langas = Tk()
sarasas = range(1, 200)

def spausdinti():
    pasirinkta = sarasas[boksas.curselection()[0]]
    uzrasas["text"] = pasirinkta

mygtukas = Button(langas, text="Spausdinti",
command=spausdinti)

uzrasas = Label(langas, text="Nieko")
boksas = Listbox(langas, selectmode=SINGLE)
boksas.insert(END, *sarasas)
boksas.pack(side=LEFT)
mygtukas.pack()
uzrasas.pack()
langas.mainloop()
```

Kaip pasiimti duomenis iš pažymėtos sąrašo vietos



```
from tkinter import *  
langas = Tk()  
  
menu = Menu(langas)  
langas.config(menu=menu)  
submenu = Menu(menu, tearoff = 0)  
  
menu.add_cascade(label="Meniu", menu=submenu)  
submenu.add_command(label="Pirmas")  
submenu.add_command(label="Antras")  
langas.mainloop()
```

Kaip sukurti meniu



```
from tkinter import *
langas = Tk()

menu = Menu(langas)
langas.config(menu=menu)
submenu = Menu(menu, tearoff = 0)

def antras():
    print("Antras!")

menu.add_cascade(label="Menu", menu=submenu)
submenu.add_command(label="Pirmas")
submenu.add_command(label="Antras",
                    command=antras)
langas.mainloop()

# Antras!
```

Kaip menu punktams priskirti funkcijas



```
from tkinter import *
langas = Tk()

menu = Menu(langas)
langas.config(menu=menu)
submenu = Menu(menu, tearoff = 0)
submenu2 = Menu(menu, tearoff = 0)
submenu3 = Menu(menu, tearoff = 0)

menu.add_cascade(label="Meniu", menu=submenu)
submenu.add_command(label="Pirmas")

submenu.add_command(label="Antras")
submenu.add_separator()
submenu.add_command(label="Trečias")

menu.add_cascade(label="Meniu 2",
menu=submenu2)
submenu2.add_command(label="1")
submenu2.add_command(label="2")

menu.add_cascade(label="Meniu 3",
menu=submenu3)

langas.mainloop()
```

**Kaip sukurti daugiau meniu,
juos atskirti**



```
from tkinter import *  
langas = Tk()  
  
status = Label(langas, text="Nieko nedaro...", bd=1, relief=SUNKEN, anchor=W)  
status.pack(side=BOTTOM, fill=X)  
langas.mainloop()
```

Statuso juostos (status bar) kūrimas



```
from tkinter import *
langas = Tk()

def daryti():
    status["text"] = "Dabar daro"

mygtukas = Button(langas, text="Daryti", command=daryti)
status = Label(langas, text="Nieko nedaro...", bd=1, relief=SUNKEN, anchor=W)
status.pack(side=BOTTOM, fill=X)

mygtukas.pack()

langas.mainloop()
```

Statuso juostos (status bar) kūrimas su mygtuku



```
status.grid(row=2, colspan=3, sticky=W+E)
```

Jei statuso juosta formuojama lentelėje



```
from tkinter import *
import webbrowser

def callback(url):
    webbrowser.open_new(url)

root = Tk()
link1 = Label(root, text="Google Hyperlink", fg="blue", cursor="hand2")
link1.pack()
link1.bind("<Button-1>", lambda e: callback("http://www.google.com"))

link2 = Label(root, text="Ecosia Hyperlink", fg="blue", cursor="hand2")
link2.pack()
link2.bind("<Button-1>", lambda e: callback("http://www.ecosia.org"))

root.mainloop()
```

Kaip sukurti veikiančią nuorodą



```
from tkinter import *
from PIL import ImageTk, Image
import os

root = Tk()
img = ImageTk.PhotoImage(Image.open("paveiksliukas.JPG"))
panel = Label(root, image = img)
panel.pack(side = "bottom", fill = "both", expand = "yes")
root.mainloop()
```

Kaip atidaryti nuotrauką



Kintamųjų naudojimas Tkinter programoje

Jei kuriant programą su Tkinter, prireiks funkcijose panaudoti kintamąjį, standartiniai kintamieji (pvz. `kintamasis = ""`) nesuveiks. Todėl patartina naudoti `StringVar`, `IntVar` kintamuosius. Jie turi `set()` (reikšmės nustatymui) ir `get()` (kintamojo reikšmės gavimui) funkcijas. Atkreipkite dėmesį, kad jos gali būti kviečiamos tik funkcijose.

```
from tkinter import *

langas = Tk()

kintamasis = StringVar("")
# kintamasis = ""

def funkcija():
    kintamasis.set("Naujas tekstas")
    print(kintamasis.get())
```

Kaip tkinter programoje padaryti kelis langus



```
import tkinter as tk

class Demo1:
    def __init__(self, master):
        self.master = master
        self.frame = tk.Frame(self.master)
        self.button1 = tk.Button(self.frame, text = 'New Window', width = 25, command = self.new_window)
        self.button1.pack()
        self.frame.pack()

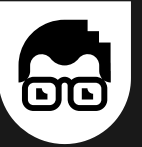
    def new_window(self):
        self.newWindow = tk.Toplevel(self.master)
        self.app = Demo2(self.newWindow)

class Demo2:
    def __init__(self, master):
        self.master = master
        self.frame = tk.Frame(self.master)
        self.quitButton = tk.Button(self.frame, text = 'Quit', width = 25, command = self.close_windows)
        self.quitButton.pack()
        self.frame.pack()

    def close_windows(self):
        self.master.destroy()

def main():
    root = tk.Tk()
    app = Demo1(root)
    root.mainloop()

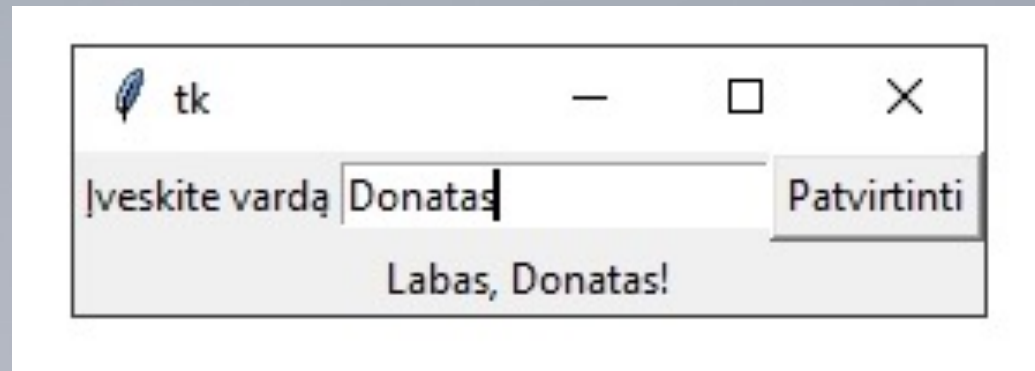
if __name__ == '__main__':
    main()
```



Užduotis nr. 1

Sukurti programą su grafine sąsaja, kuri:

- Turėtų laukelį su užrašu "Įveskite vardą", kuriame vartotojas galėtų įvesti vardą
- Turėtų mygtuką su užrašu "Patvirtinti", kurį nuspaudus, programa po lauku atspausdintų "Labas, {vardas}!"





Užduotis nr. 2

Patobulinti 1 užduoties programą, kad ji:

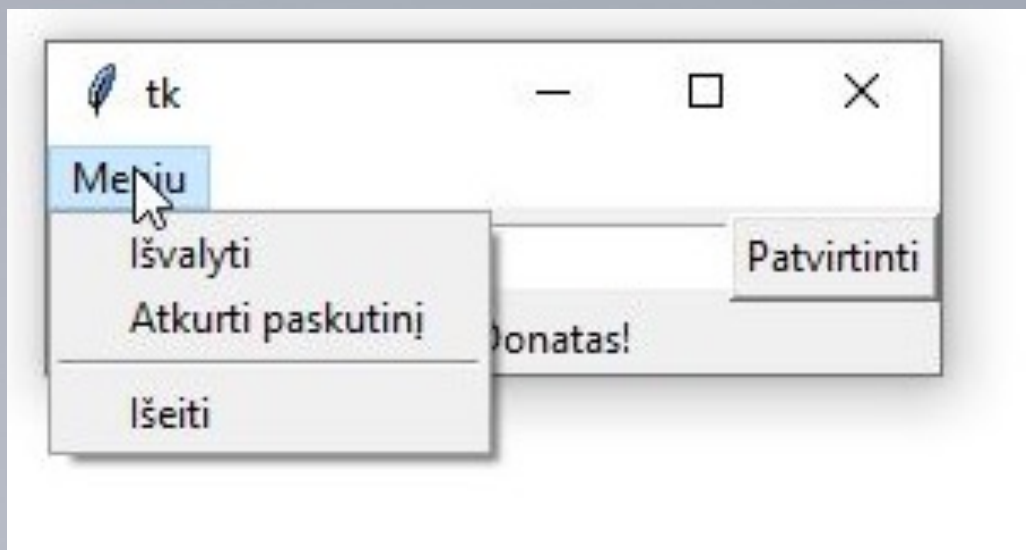
- Atspausdintų pasisveikinimą ne tik nuspaudus mygtuką, bet ir paspaudus mygtuką "Enter"



Užduotis nr. 3

Patobulinti 2 užduoties programą, kad ji turėtų meniu pavadinimu "Meniu", kuriame:

- Būtų punktas "Išvalyti", kurį paspaudus išsitrintų tekstas eilutėje, kurioje spausdinamas pasisveikinimo tekstas
- Būtų punktas "Atkurti", kurį paspaudus pasisveikinimo teksto eilutėje būtų atspausdintas paskutinis atspausdintas tekstas
- Būtų punktas "Išeiti", kurį paspaudus užsidarytų programos langas
- Tarp menių punktų "Atkurti" ir "Išeiti" būtų atskyrimo brūkšnys

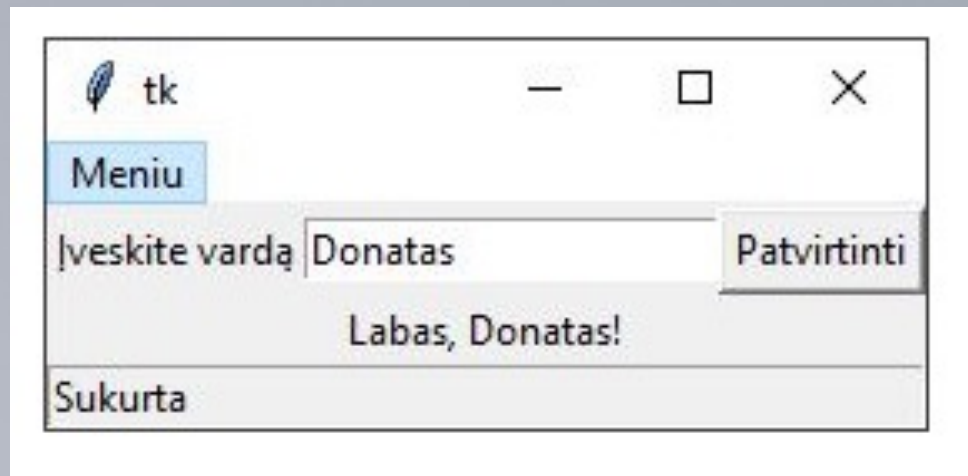




Užduotis nr. 4

Patobulinti 3 užduoties programą, kad ji turėtų statuso juostą apačioje, kurioje:

- Būtų rodoma "Sukurta", kai atspausdinamas pasisveikinimo tekstas
- Būtų rodoma "Išvalyta", kai ištrinamas pasisveikinimo tekstas
- Būtų rodoma "Atkurta", kai atkuriamas paskutinis pasisveikinimo tekstas Nuspaudus klaviatūros mygtuką "Escape", uždarytą programos langą





Namų darbas

Užbaigti klasėje nepadarytas užduotis



Išspręsti paskaitos uždaviniai (įkelti ketvirtadienį)

<https://github.com/aurimas13/Python-Beginner-Course/tree/main/Programs>

Tkinter button names

<https://web.archive.org/>

Tkinter bibliotekos mygtukų kodai

Tkinter info

https://www.tutorialspoint.com/python/python_gui_programming.htm

Visa informacija apie Tkinter biblioteka

**Naudinga
informacija**