Tema 1.
Componente de bază Tkinter

Ce ne așteaptă?

1. Introducere în modulul Tkinter
2. Componenta - Label
3. Componenta - Button
4. Componenta - Entry

1. Ce reprezintă programarea GUI?
2. Cum se creează fereastra de bază în Tkinter?
3. Cum se setează parametrii ferestrei de bază în Tkinter?
4. Care sunt etapele introducerii componentelor pe fereastra de bază?
5. Ce reprezintă componenta Label și cum se creează?
6. Care sunt parametrii componentei Label?
7. Ce reprezintă componenta Button și cum se creează?
8. Cum se atribuie o acțiune componentei Button?
9. Cum componenta Button se decorează cu o imagine?
10. Ce reprezintă componenta Entry și cum se creează?
11. Cum se citește, înscrie și șterge conținutul componentei Entry?

1. Introducere în modulul Tkinter

Introducere în programarea GUI

- **Programarea GUI presupune elaborare unor aplicații cu interfață grafică de utilizator**

  - utilizează ferestre ce conțin butoane, bare de introducere a textului, texte informative si altele

  - utilizatorul utilizează mouse-ul pentru a interacționa cu componentele ferestrei

- **Module Python specializate în programarea GUI**

  - Tkinter,
  - PyQt5,
  - Kivy,
  - wxPython,
  - PySimpleGUI

Crearea ferestrei de bază

- **Modulul Tkinter – modul încorporat în Python ce permite elaborarea aplicațiilor GUI cu design dependent de sistemul de operare**

- **Importul componentelor modulului TKinter**

```
from tkinter import *
```

- **Crearea ferestrei de bază – toplever widget – clasa *Tk()***

```
root = Tk()
```

- **Afișarea în continuu a ferestrei – metoda *mainloop()* –** întotdeauna se află la sfârșitul programului

```
root.mainloop()
```

Setarea titlului și dimensiunii ferestrei de bază

- **Modificarea denumirii ferestrei – metoda *title()***

```
root.title('Prima aplicatie GUI')
```

- **Setarea dimensiunilor ferestrei – metoda *geometry('latime_pixelixînaltime_pixeli')***

```
root.geometry('600x400')
```

- **Blocarea redimensionarii ferestrei – metoda *resizable(latime_bool, înaltime_bool)***

```
root.resizable(False, True) # blocare redimensionare pe lățime
root.resizable(True, False) # blocare redimensionare pe înălțime
root.resizable(False, False) # blocare redimensionare totală
```

- **Fixarea dimensiunilor minime la redimensionare – metoda minsize(*latime_min*, *înaltime_min*)**

```
root.minsize(100, 50)
```

- **Fixarea dimensiunilor maxime la redimensionare – metoda minsize(*latime_max*, *înaltime_max*)**

```
root.maxsize(900, 700)
```

Setarea icon și culorii ferestrei de bază

- **Modificarea icon ferestrei – metoda *iconbitmap('calea_fisier.ico')***

```
root.iconbitmap('icon.ico') # necesită prezenta icon.ico in forderul curent
```

- **Setarea transparentei ferestrei – metoda *attributes('-alpha', grad_transparenta)***

```
root.attributes('-alpha', 0.5)
```

- **Setarea culorii ferestrei – metoda *config(bg='culoare')***

```
root.config(bg='green')
```

  - *Selectarea culorii dupa denumire în engleza: green, red, blue*
  - *Selectarea culorii conform codului hexazecimal: #4567AF*

Crearea și amplasare componentelor

- **Etapele de amplasare a componentelor**
  - Crearea și parametrizarea componentei

    nume_variabila = Clasa_Componentă(componenta_parinte, parametri)

    ```
    label = Label(root, text='Salutare tuturor')
    ```
  - Amplasarea componentei pe fereastra părinte

    nume_variabila.metodă_amplasare(parametri)

    ```
    label.pack()
    ```
- **Fixarea și citirea parametrilor**
  - Fixarea parametrilor în constructorul clasei componentei

    ```
    label = Label(root, bg='green')
    ```
  - Fixarea parametrilor cu ajutorul metodei config()

    ```
    label.config(bg='green')
    ```
  - Citirea valorii parametrului cu ajutorul metodei cget()

    ```
    print(label.cget('text'))
    ```

2. Componenta - Label

Crearea și amplasarea componentei - Label

- Se utilizează pentru introducerea unui text ce nu poate fi modificat
- Se creează utilizând clasa Label()

```
label = Label(root, text='Salutare tuturor')
```

- Amplasarea componentei

```
label.pack()
```

Parametrii ai componentei Label (1)

- relief – modeul de prezentarea a hotarului componentei: FLAT, RAISED, SUNKEN GROOVE sau RIDGE

```
label.config(relief=RAISED)
```

- bd – lățimea hotarului (în pixeli)

```
label.config(bd=10)
```

- height – înalțimea componentei (în înălțimi a caracterelor)

```
label.config(height=3)
```

- width – lațimea componentei (în caractere)

```
label.config(width=64)
```

- pady – spațiu dintre hotare și text pe verticală (în pixeli)

```
label.config(pady=3)
```

Parametrii ai componentei Label (2)

- padx – spațiu dintre hotare și text pe orizontală (în pixeli)

```
label.config(padx=10)
```

- anchor – poziționarea textului pe componentă: CENTER, N, NE, E, SE, S, SW, W, NW)

```
label.config(anchor=W)
```

- font – selectarea fontului și înălțimii textului cu opțiunile de bold și italic

```
label.config(font="Tahoma 20 italic bold")
```

- bg – selectarea culorii de fon a componentei

```
label.config(bg="red")
```

- fg – selectarea culorii textului

```
label.config(fg="yellow")
```

3. Componenta - Button

Crearea și amplasarea componentei - Button

- Se utilizează pentru realizarea unor acțiuni
- Se creează utilizând clasa Button()

```
buton = Button(root, text='Click aici')
```

- Amplasarea componentei

```
buton.pack()
```

Parametrii componentei Button

- Componenta Button acceptă toți parametrii specificați la componenta Label
- state – permite activarea sau dezactivarea butonului: NORMAL, DISABLED

```
buton.config(state=DISABLED)
```

- command – specifică funcția ce se va executa la apăsarea butonului

```
buton = Button(root, text='Click aici', command=afisare)
```

- Exemplu de utilizare a parametrului command

```
def afisare():
    print("Salutare tuturor")

buton = Button(root, text='Click aici', command=afisare)
buton.pack()
```

Imaginea ca buton

- Se creează un obiect al clasei PhotoImage

```
imagine = PhotoImage(file='imagine.png')
```

- Buton doar din imagine – parametrul text se exclude iar parametrului image a butonului i se atribuie obiectul PhotoImage creat

```
buton = Button(root, image=imagine)
```

- Buton din text și imagine – conține parametrii text, image și compound. Parametrul compound specifică poziția imaginii LEFT, RIGHT, TOP, BOTTOM

```
buton = Button(root, text="Modifica", image=imagine, compound=LEFT)
```

Exemplu de utilizare a componentelor Label și Button

```
from tkinter import *
root = Tk()
root.title('Schimbare culoare text’)
root.geometry('300x150’)

def schimba():
    if label.cget("text")== "Text colorat rosu":
        label.config(text="Text colorat verde")
        label.config(fg="green")
    else:
        label.config(text="Text colorat rosu")
        label.config(fg="red")

label=Label(root, text="Text colorat rosu", fg="red")
label.pack()
imagine = PhotoImage(file='imagine.png')
buton = Button(root, text="Schimba culoare", image=imagine,
compound=TOP,  command = schimba))
buton.pack()
root.mainloop()
```

4. Componenta - Entry

Crearea și amplasarea componentei - Entry

- Se utilizează pentru a permite utilizatorului să introducă text pe o singură linie
- Se creează utilizând clasa Entry()

```
intrare = Entry(root)
```

- Amplasarea componentei

```
intrare.pack()
```

Citirea conținutului Entry

- Componenta Entry acceptă toți parametrii specificați la componenta Label
- Parametrul show – permite ascunderea textului si afișarea doar a caracterului specificat

```
intrare = Entry(root, show='*')
```

- Metoda get() - citește conținutul componentei și îl returnează sub formă de string

data= nume_variabila.get()

- Exemplu de utilizare

```
def afisare():
    continut = intrare.get()
    print(continut)

intrare = Entry(root, width=30)
intrare.pack()

buton = Button(root, text="afisare", command = afisare)
buton.pack()
```

Înscriere conținut în Entry

- **Metoda insert() permite înscrierea programabilă a conținutului în Entry:**

  ***nume_variabila*.insert(*index_inceput*, *continut*)**

- **Pentru înscrierea unui conținut nou la sfârșitul celui existent index_inceput va fi constanta END:**

- **Exemplu de înscriere la început**

```
def inscriere():
    intrare.insert(0,text)

intrare = Entry(root, width=30)
intrare.pack()

buton = Button(root, text="inscrie", command = inscriere)
buton.pack()

text = "Salut "
```

Ștergerea conținutului Entry

- Metoda delete() – permite ștergerea programabilă a conținutului Entry

  nume_variabila.delete(*index_inceput, index_sfârsit*)

- Pentru ștergerea întregului conținut index_inceput va fi 0 iar index_sfarsit - constanta END:

- Exemplu de ștergere conținut

```
def stergere():
    intrare.delete(0,END)

intrare = Entry(root, width=30)
intrare.pack()

buton = Button(root, text="stergere", command = stergere)
buton.pack()
```

Exemplu de utilizare a componentelor Label, Button și Entry

```
from tkinter import *

root = Tk()
root.title('Logare’)

def logare():

    if intrare1.get()=="1234" and intrare2.get()=="1111":
        label3.config(text="Logare cu succes!")
        label3.config(fg="green")

    else:
        label3.config(text="Logare esuata!")
        label3.config(fg="red")

    label3.pack()

    intrare1.delete(0,END)
    intrare2.delete(0, END)
```

Exemplu de utilizare a componentelor Label, Button și Entry (continuare)

```
label1=Label(root, text="Introduceti adresa de email:", width=44, anchor=W)
label1.pack()

intrare1 = Entry(root, width=50)
intrare1.pack()

label2=Label(root, text="Introduceti parola:", width=44, anchor=W)
label2.pack()

intrare2 = Entry(root, width=50, show='*')
intrare2.pack()

buton = Button(root, text="Logare", command = logare)
buton.pack()

label3=Label(root)

root.mainloop()
```