INFORMATICĂ · PYTHON · CLASA A IX-A · MEMORAREA DATELOR ȘI ORGANIZAREA CODULUI · LECȚIA 11

Interfețe grafice cu Tkinter (1): fereastră, etichete, butoane, casete text

Durată: 100 min Nivel: mate-info · Python Înainte: lecțiile py9-10

CE VEI ȘTI DUPĂ LECȚIE

- Construiești o fereastră cu Tk, Label, Button și Entry, așezate cu pack().
- Legi un buton de o funcție prin command și explici de ce funcția se scrie fără paranteze.
- Citești textul dintr-o casetă cu get(), îl convertești și afișezi rezultatul într-o etichetă sau cu messagebox.
- Separi calculul (o funcție testabilă) de interfață și verifici datele introduse de utilizator.

1. De la consolă la fereastră

Un program de consolă merge „de sus în jos”: citește, calculează, afișează, se termină. Un program cu **interfață grafică** desenează o fereastră și apoi **așteaptă evenimente** (un clic, o tastă). La fiecare eveniment rulează o funcție pe care ai legat-o tu de el.

```
import tkinter as tk

fereastra = tk.Tk()                        # obiectul fereastră principală
fereastra.title("Primul program")
fereastra.geometry("300x120")              # lățime x înălțime, în pixeli

eticheta = tk.Label(fereastra, text="Salut, clasa a IX-a!", font=("Arial", 14))
eticheta.pack(pady=20)                     # o așază în fereastră

fereastra.mainloop()                       # pornește bucla de evenimente
```

Se deschide o fereastră cu titlul „Primul program” și textul „Salut, clasa a IX-a!”. Programul se termină abia când închizi fereastra.

	Consolă	Interfață grafică
intrare	`input()`	o casetă `Entry`, citită cu `get()`
ieșire	`print()`	o etichetă `Label` sau o fereastră `messagebox`
ordinea acțiunilor	stabilită de program	aleasă de utilizator (ce buton apasă, când)
sfârșitul programului	după ultima instrucțiune	când se închide fereastra

2. Elementele grafice sunt obiecte

`Tk`, `Label`, `Button`, `Entry` sunt **clase** din biblioteca `tkinter` (vezi lecția py9-10). Fiecare element (numit *widget*) se creează prin instanțiere; primul argument este **părintele** (fereastra în care stă), apoi urmează opțiuni cu nume:

Clasa	Ce este	Opțiuni des folosite
`tk.Tk()`	fereastra principală (una singură)	metodele `title(...)`, `geometry("LxÎ")`, `mainloop()`
`tk.Label(p, ...)`	etichetă: text afișat	`text`, `font`, `fg` (culoarea textului), `bg` (fundal)
`tk.Button(p, ...)`	buton	`text`, `command` (funcția apelată la clic), `width`
`tk.Entry(p, ...)`	casetă pentru un rând de text	`width`

Un widget creat **nu apare** până nu îl așezi cu `pack()` (sau cu `grid()` / `place()`, în lecția py9-12). `pack()` le pune unul sub altul, în ordinea apelurilor; `padx` și `pady` lasă spațiu în jur.

Opțiunile se pot schimba oricând: `eticheta.config(text="Alt text")` sau `eticheta["text"] = "Alt text"`. Valoarea curentă o afli cu `eticheta.cget("text")`.

3. Butonul și evenimentul command

```
import tkinter as tk

apasari = 0

def la_click():
    global apasari                  # modificăm variabila globală
    apasari += 1
    eticheta.config(text=f"Ai apăsat de {apasari} ori")

fereastra = tk.Tk()
fereastra.title("Contor")
eticheta = tk.Label(fereastra, text="Încă nu ai apăsat")
eticheta.pack()
buton = tk.Button(fereastra, text="Apasă-mă", command=la_click)
buton.pack()
fereastra.mainloop()
```

La pornire eticheta arată „Încă nu ai apăsat"; după trei clicuri arată „Ai apăsat de 3 ori".

> **ATENȚIE — GREȘELI FRECVENTE**
>
> Scrii `command=la_click`, **fără paranteze**: îi dai butonului funcția, ca s-o apeleze el la fiecare clic. Cu `command=la_click()` funcția rulează **o singură dată**, chiar la crearea butonului, iar butonul primește rezultatul ei (`None`) și nu mai face nimic la clic.

Funcția legată de `command` nu are parametri și nu întoarce nimic util: își face efectul modificând widget-uri sau variabile.

4. Caseta Entry și metoda get()

```
def saluta():
    nume = caseta.get()                          # textul din casetă
    rezultat.config(text=f"Bună, {nume.strip().capitalize()}!")
    caseta.delete(0, tk.END)                     # golește caseta

fereastra = tk.Tk()
tk.Label(fereastra, text="Cum te cheamă?").pack()
caseta = tk.Entry(fereastra, width=20)
caseta.pack()
tk.Button(fereastra, text="Salută", command=saluta).pack()
rezultat = tk.Label(fereastra, text="")
rezultat.pack()
fereastra.mainloop()
```

Dacă scrii `maria` și apeși butonul, eticheta devine `Bună, Maria!`, iar caseta se golește.

Metoda `Entry`	Efect
`caseta.get()`	întoarce textul din casetă (**mereu** `str`)
`caseta.delete(0, tk.END)`	șterge tot textul
`caseta.insert(0, "text")`	inserează textul la început (`tk.END`: la sfârșit)

ATENȚIE — GREȘELI FRECVENTE

`get()` se comportă ca `input()`: dă **text**. Dacă în casetă este `5`, `caseta.get() + 1` dă `TypeError`; corect este `int(caseta.get()) + 1`.

Un widget pe care nu îl mai folosești (eticheta „Cum te cheamă?") poate fi creat și așezat direct: `tk.Label(...).pack()`. Pe cele folosite în funcții le păstrezi în variabile (vezi secțiunea 7).

5. Ferestre de mesaj: messagebox

Modulul `tkinter.messagebox` (importat separat: `from tkinter import messagebox`) deschide ferestre mici, standard, cu un titlu și un mesaj: `showinfo(titlu, mesaj)` pentru informații, `showwarning(...)` pentru avertismente, `showerror(...)` pentru erori. `askyesno(titlu, întrebare)` afișează butoanele Da / Nu și întoarce `True` sau `False`.

6. O aplicație completă: câți divizori are un număr?

Regula de aur: **calculul stă într-o funcție obișnuită**, fără nimic grafic. Funcția legată de buton doar citește, verifică, apelează calculul și afișează.

```python
import tkinter as tk
from tkinter import messagebox

def nr_divizori(n):                        # LOGICA: testabilă fără fereastră
    c = 0
    d = 1
    while d * d <= n:
        if n % d == 0:
            c += 1 if d * d == n else 2
        d += 1
    return c

def calculeaza():                          # INTERFAȚA: citește, verifică, afișează
    text = caseta.get().strip()
    if not text.isdigit() or int(text) == 0:
        messagebox.showerror("Eroare", "Scrie un număr natural nenul.")
        return
    n = int(text)
    messagebox.showinfo("Rezultat", f"{n} are {nr_divizori(n)} divizori.")

fereastra = tk.Tk()
fereastra.title("Divizori")
tk.Label(fereastra, text="Număr natural:").pack(padx=10, pady=5)
caseta = tk.Entry(fereastra)
caseta.pack(padx=10)
tk.Button(fereastra, text="Câți divizori?", command=calculeaza).pack(pady=10)
fereastra.mainloop()
```

Testarea se face în două straturi:

Ce testezi	Cum	Rezultat obținut
logica	`for n in [1, 12, 36, 97, 100]: print(nr_divizori(n), end=" ")` în consolă	`1 6 9 2 9`
interfața, caz normal	scrii `12`, apeși butonul	mesaj „12 are 6 divizori.”
interfața, cu spații	`36`	„36 are 9 divizori.” (`strip()` le elimină)
interfața, date greșite	`abc`, `0`, `-5`, casetă goală	mesajul de eroare

`isdigit()` este adevărat doar pentru un text nevid format numai din cifre, deci respinge și `-5`, și caseta goală.

SFAT

Pentru numere reale sau negative, unde `isdigit()` nu ajută, Python are construcția `try` / `except`: încearcă conversia și, dacă dă eroare, execută altceva.

```python
try:
    x = float(caseta.get())
except ValueError:
    messagebox.showerror("Eroare", "Nu este un număr.")
```

7. Capcane frecvente

Ce scrii	Ce se întâmplă	Corect
`command=calculeaza()`	funcția rulează o dată, la pornire; butonul nu mai face nimic	`command=calculeaza`
uiți `pack()`	widget-ul există, dar nu apare	`w.pack()`
uiți `mainloop()`	programul se termină imediat și fereastra dispare (în unele medii, ca IDLE, poate rămâne deschisă)	`fereastra.mainloop()` la final
`x = caseta.get() + 1`	`TypeError`	`int(caseta.get()) + 1`
`caseta = tk.Entry(f).pack()`	`caseta` devine `None` (rezultatul lui `pack()`); `caseta.get()` dă `AttributeError`	`caseta = tk.Entry(f)`, apoi `caseta.pack()`
`input()` într-o aplicație grafică	programul așteaptă în consolă, fereastra „îngheață”	o casetă `Entry`

8. Verifică-te

1. Care este rolul lui `mainloop()`? Ce s-ar întâmpla cu un `print("gata")` scris după el?
2. De ce în `tk.Label(fereastra, text="…")` apare `fereastra` ca prim argument?
3. Caseta este goală. Ce conține după `caseta.insert(0, "lume")` și apoi `caseta.insert(0, "salut ")`?
4. Ce tip are valoarea întoarsă de `get()`? Ce scrii ca să obții un număr real?
5. Un buton are `command=afiseaza()`. Ce observi la rulare?
6. De ce scriem `nr_divizori` separat, în loc să punem tot calculul în `calculeaza`?

Răspunsuri

1. Pornește bucla de evenimente: ține fereastra deschisă și apelează funcțiile legate de butoane. `print("gata")` se execută abia după închiderea ferestrei.
2. Este părintele: fereastra în care va fi pusă eticheta.
3. `salut lume` (al doilea text este inserat la început, înaintea primului).
4. `str`. Scrii `float(caseta.get())`.
5. `afiseaza` se execută o dată, la pornire, iar clicul pe buton nu mai face nimic.
6. Funcția de calcul se poate testa rapid, pe multe valori, fără să deschizi fereastra; aceeași funcție poate fi folosită și într-un program de consolă.