FIȘĂ DE LUCRU · PYTHON · CLASA A IX-A · FIȘA 12 · LECȚIA PY9-12

Interfețe grafice cu Tkinter (2): grid, place, Frame, Text, Canvas

Durată: 100 min Nivel: mate-info · Python

CE VEI ȘTI DUPĂ LECȚIE

- Prezici așezarea cu grid, conținutul unui Text după operații cu indici și coordonatele figurilor de pe Canvas.
- Repari conflictele pack/grid, indicii greșiți și coordonatele inversate.
- Construiești aplicații cu formulare, desene și animații simple, cu logica testată separat.

Ce trebuie să știi (o pagină)

Ce vrei	Cum scrii
așezare în tabel	`w.grid(row=r, column=c)`; opțiuni: `columnspan=2`, `sticky="e"` / `"we"`, `padx`, `pady`
așezare la coordonate	`w.place(x=…, y=…)` (pixeli, din colțul stânga-sus)
grup de elemente	`fr = tk.Frame(f)`; elementele din el au părintele `fr`
regula de aur	într-un container, **o singură** metodă de așezare: `pack` **sau** `grid`
casetă cu mai multe rânduri	`t = tk.Text(f, width=40, height=6)`
indici `Text`	`"linie.coloană"` (linii de la 1, coloane de la 0), `"2.end"`, `"end"`
citești / scrii / ștergi	`t.get("1.0", "end")` (are `\n` la final) · `t.insert("end", "…")` · `t.delete("1.0", "end")`
pânză	`c = tk.Canvas(f, width=…, height=…, bg="white")`; `(0, 0)` stânga-sus, `y` crește **în jos**
figuri	`create_line(x1, y1, x2, y2)`, `create_rectangle(x1, y1, x2, y2, fill=…)`, `create_oval(…)` (în dreptunghiul dat), `create_text(x, y, text=…)`
cerc cu centrul (x, y) și raza r	`c.create_oval(x - r, y - r, x + r, y + r)`
lucrezi cu o figură	`id = c.create_…(…)` · `c.coords(id)` · `c.coords(id, x1, y1, x2, y2)` · `c.move(id, dx, dy)` · `c.delete(id)` / `c.delete("all")`

Ce vrei	Cum scrii
acțiune întârziată	`f.after(ms, functie)`: apelează `functie` o dată, după `ms` milisecunde
tastă apăsată	`w.bind("<Return>", functie)`: `functie(event)` (scrie `event=None` ca s-o poți apela și din buton)
listă vizuală, meniu	`tk.Listbox`: `insert("end", t)`, `curselection()`, `delete(i)`, `size()`; `tk.Menu`: `add_cascade`, `add_command(label=…, command=…)`
rotunjire „de la ,50 în sus”	`int(m + 0.5)` pentru `m ≥ 0`; `round()` rotunjește jumătățile spre par

SFAT

Înainte să scrii o interfață, desenează-o pe hârtie ca un tabel: ce stă pe fiecare rând și în fiecare coloană. Codul cu `grid` se scrie apoi aproape singur.

Nivelul 1 · Prezice

★ citește și prezice

P1. Desenează (ca tabel) cum sunt așezate etichetele. Care stă singură pe un rând, centrată sub celelalte?

```
tk.Label(f, text="A").grid(row=0, column=0)
tk.Label(f, text="B").grid(row=0, column=1)
tk.Label(f, text="C").grid(row=1, column=1)
tk.Label(f, text="D").grid(row=2, column=0, columnspan=2)
```

P2. Caseta `t` conține trei rânduri: `linia unu`, `linia doi`, `linia trei`. Ce întorc?

```
t.get("2.0", "2.5")
t.get("3.6", "end")
t.get("1.0", "1.end").upper()
t.get("1.6", "2.2")
```

P3. Ce lățime și ce înălțime are dreptunghiul? Ce afișează `print(c.coords(d))` la final?

```
d = c.create_rectangle(20, 30, 70, 90)
c.move(d, 10, -10)
```

P4. Ce afișează? (Numerele de identificare nu se refolosesc.)

```
a = c.create_line(0, 0, 10, 10)
b = c.create_oval(0, 0, 10, 10)
x = c.create_text(5, 5, text="x")
c.delete(b)
n = c.create_rectangle(1, 1, 2, 2)
print(n, c.find_all())          # find_all() dă figurile rămase
```

P5. Caseta `t` este goală. După `t.insert("end", "3 4\n5\n")`, ce valoare are `s = t.get("1.0", "end")`? Ce afișează `print(sum(int(v) for v in s.split()))`?

P6. Ce afișează? Care dintre valori te-ar încurca la calculul mediei unui elev?

```
print(round(2.5), round(3.5), int(2.5 + 0.5), round(7.25, 1))
```

Nivelul 2 · Găsește și repară

★★ găsește și repară

P7. Programul se oprește cu `TclError: cannot use geometry manager "grid" inside "." …`. Explică de ce și repară-l în **două** feluri.

```
tk.Label(f, text="Titlu").pack()
tk.Entry(f).grid(row=1, column=0)
```

P8. Pentru caseta `Text` numită `t`, elevul a încercat `t.get()` și apoi `t.get(0, "end")`. Prima dă `TypeError`, a doua `TclError: bad text index "0"`. Scrie apelul corect pentru tot textul, fără `\n`-ul final.

P9. Trebuia un cerc cu centrul `(100, 50)` și raza 30. Ce s-a desenat de fapt? Corectează.

```
c.create_oval(100, 50, 30, 30)
```

P10. Pe un `Canvas` cu înălțimea 120, barele graficului „atârnă" de marginea de sus în loc să stea pe cea de jos. Repară rândul.

```
c.create_rectangle(x, 0, x + 20, nota * 10, fill="steelblue")
```

P11. *Completează.* Butoanele cu cifrele 1–9 trebuie să formeze o tastatură 3 × 3 (1 2 3 pe primul rând, 4 5 6 pe al doilea, 7 8 9 pe al treilea), iar `0` să stea dedesubt, pe toată lățimea.

```
for d in range(1, 10):
    tk.Button(f, text=d, width=4).grid(row=________, column=________)
tk.Button(f, text=0, width=4).grid(row=3, column=0, ________, sticky="we")
```

P12. *Rânduri amestecate.* O casetă sus și, sub ea, butoanele `OK` și `Renunță` alăturate. Un rând este în plus: care și ce eroare ar da?

```
A)  tk.Button(butoane, text="Renunță").pack(side="left")
B)  caseta.grid(row=0, column=0)
C)  butoane = tk.Frame(f)
D)  butoane.pack()
E)  caseta = tk.Entry(f)
F)  tk.Button(butoane, text="OK").pack(side="left")
G)  butoane.grid(row=1, column=0)
```

Nivelul 3 · Construiește

★★ scrie cod

Fiecare aplicație: logica în funcții fără elemente grafice, testată pe **toate** cazurile din tabel.

P13. Convertor de unități. O casetă, trei butoane alăturate într-un `Frame` (`km → mi`, `mi → km`, `kg → lb`) și o etichetă cu rezultatul, cu 3 zecimale. 1 milă = 1,609344 km; 1 livră = 0,45359237 kg.

valoare	buton	rezultat
`10`	km → mi	`6.214 mi`
`26.2`	mi → km	`42.165 km`
`1`	kg → lb	`2.205 lb`
`-5`	mi → km	`-8.047 km`
`abc`	oricare	mesaj de eroare, cu roșu

P14. Statistica unui text. O casetă `Text`, un buton „Analizează” și o etichetă `rânduri: R, cuvinte: C, caractere: K` (caracterele fără spații și fără sfârșituri de rând; se numără și rândurile goale dintre cele scrise).

text	rezultat
`Ana are mere`	`rânduri: 1, cuvinte: 3, caractere: 10`
`Ana are mere` / `Ion are pere`	`rânduri: 2, cuvinte: 6, caractere: 20`
(gol)	`rânduri: 0, cuvinte: 0, caractere: 0`
`unu doi` / (rând gol) / `trei`	`rânduri: 3, cuvinte: 3, caractere: 10`

P15. Ținta. Pe un `Canvas` 220 × 220, desenează 5 cercuri cu centrul `(110, 110)` și razele 100, 80, 60, 40, 20, alternând roșu și alb (cel mare roșu). Verificare: `c.coords(...)` pentru cercul mare dă `[10.0, 10.0, 210.0, 210.0]`, iar pentru cel mic `[90.0, 90.0, 130.0, 130.0]`.

P16. Tabla de șah. Desenează o tablă 8 × 8 cu pătrate de 40 de pixeli (pătratul din stânga-sus alb). Verificare: 64 de dreptunghiuri, 32 închise la culoare, ultimul pătrat (dreapta-jos) are coordonatele `[280.0, 280.0, 320.0, 320.0]` și este alb.

P17. Lista de cumpărături. O casetă `Entry` și butonul „Adaugă” pe primul rând (`grid`), dedesubt o casetă `Text` și o etichetă `k produse`. Fiecare produs se adaugă pe un rând nou în `Text`; caseta se golește; textul gol sau doar cu spații este ignorat.

se adaugă, în ordine	eticheta	conținutul `Text`
`lapte`, ` `, `pâine`, `ouă`	`3 produse`	`lapte`, `pâine`, `ouă`, fiecare pe un rând

P18. Bila. Un `Canvas` 200 × 200 cu o bilă de rază 10 care pornește din centru și patru butoane (↑ ← ↓ →) așezate cu `grid` într-un `Frame`. Fiecare clic mută bila 10 pixeli, dar ea nu iese de pe pânză. Logica: `muta(x, y, dx, dy)` întoarce noul centru.

apel / clicuri	rezultat
`muta(100, 100, 10, 0)`	`(110, 100)`
`muta(190, 100, 10, 0)`	`(190, 100)`
`muta(10, 10, -10, -10)`	`(10, 10)`

apel / clicuri	rezultat
de 12 ori →, apoi de 3 ori ↑	`c.coords(bila)` = `[180.0, 60.0, 200.0, 80.0]`

Nivelul 4 · Pune mintea la contribuție

★★★ pune mintea la contribuție

P19. Care metodă? Un formular are 4 etichete și 4 casete. Profesorul mărește fontul tuturor etichetelor de la 10 la 20. Ce se întâmplă dacă formularul a fost așezat cu `place` și ce se întâmplă dacă a fost așezat cu `grid`? Ce alegi pentru un calculator de buzunar și ce pentru un joc cu o navă care se mișcă?

P20. Problemă inversă. Scrie apelurile `grid` pentru un mini-calculator cu această așezare: pe rândul 0, o casetă pe toată lățimea; pe rândul 1, butoanele `+`, `−`, `×`, `÷`; pe rândul 2, butonul `=` pe trei coloane și `C` în ultima.

P21. Din matematică pe pânză. Pe hârtie, axa `Oy` urcă; pe `Canvas` coboară. Scrie funcția `la_canvas(x, y, H)` care transformă un punct din sistemul de la matematică (originea jos-stânga) într-un punct pe o pânză cu înălțimea `H`. Teste: `la_canvas(0, 0, 100)` → `(0, 100)`, `la_canvas(30, 90, 100)` → `(30, 10)`. Folosește-o ca să desenezi, cu linii scurte, graficul lui `y = x²/10` pentru `x` de la 0 la 30 pe o pânză 100 × 100 (un pixel pe unitate pe Ox, unul pe unitate pe Oy). Ce valoare are `y` pentru `x = 30`? Încape?

P22. Rotunjirea la negative. `int(m + 0.5)` este corect pentru medii, dar nu pentru orice număr. Calculează `int(-2.5 + 0.5)` și `int(-2.7 + 0.5)`. Care rezultat este greșit față de „cel mai apropiat întreg”? De ce aici nu ne deranjează?

P23. Mini-proiect: Start și Stop. Fereastră cu fundal colorat, o casetă pentru primul termen, o etichetă mare și două butoane, `Start` și `Stop`. După `Start`, la fiecare 0,7 secunde eticheta afișează următorul termen al șirului lui Collatz (`n / 2` dacă `n` este par, `3n + 1` dacă este impar), până la 1. `Stop` oprește afișarea. Folosește `f.after(700, pas)`, unde funcția `pas` se programează singură din nou. Test: pornind de la 6, se afișează pe rând `6 3 10 5 16 8 4 2 1`, apoi afișarea se oprește.

P24. Enter și Escape. O casetă `Entry` și o etichetă. La **Enter** eticheta arată textul din casetă cu majuscule (fără spațiile de la margini), sau `scrie ceva` dacă textul e gol. La **Escape** se golesc și caseta, și eticheta. Scrie funcțiile și cele două apeluri `bind`. Ce se întâmplă la Enter dacă funcția nu are parametrul `event`?

P25. Lista de participanți. O casetă și un `Listbox`. Enter adaugă numele din casetă (nu și textele goale) și golește caseta. Meniul **Acțiuni** are trei comenzi: „Numără” (un `messagebox` cu `3 participanți`), „Sortează” (reface lista în ordine alfabetică) și „Șterge selecția”. Scrie funcțiile și construiește meniul. Ce întoarce `curselection()` când nu e nimic ales?
