test-e.ro

CAIET ELEV

Lecția 8

Modalități de comunicare cu programul

*Firul lecției de azi: un program nu vorbește cu lumea doar prin tastatură. Aceeași informație poate circula pe **trei drumuri** diferite — prin **consolă** (rânduri de text), printr-o **fereastră grafică** (butoane și casete) sau printr-un **fișier** (date care rămân pe disc). Se schimbă doar cum arată dialogul, nu ce face programul.*

DISCIPLINA Informatică **CLASA** a IX-a · mate-info **LIMBAJ** Python 3

NUMELE ELEVULUI .....

CUM FOLOSEȘTI CAIETUL

Acest caiet adună exercițiile scrise din Lecția 8. Rezolvă-le pe hârtie — recunoști cine „vorbește" într-un dialog, alegi drumul de comunicare potrivit pentru fiecare situație, prezici ce afișează consola, identifici piesele unei ferestre și deosebești cele trei modalități. **Fereastra grafică (Tkinter) și fișierele apar doar ca recunoaștere** — le vei programa efectiv mai târziu în acest an. Singurul drum pe care îl scrii acum este **consola**, cu `input()` și `print()`. Folosește doar ce s-a predat până acum (lecțiile 1–7). Verificarea automată a codului rămâne pe platformă.

CUPRINS

1 Cum vorbește programul cu noi .....
2 Consolă, fereastră, fișier .....
3 Consola, în detaliu .....
4 Interfața grafică .....
5 Fișierul · memoria de durată .....
6 Detectivul interfețelor .....
7 Programează dialogul prin consolă .....

SECȚIUNEA 1 · RECAPITULARE

1 Cum *vorbește* programul cu noi

Până acum, toate programele tale au comunicat într-un singur fel: prin **consolă**. Tu scrii ceva la tastatură, programul răspunde cu text pe ecran. Pare puțin, dar e o conversație completă: programul **cere** date (`input()`) și **livrează** rezultate (`print()`). Una pentru intrare, una pentru ieșire.

DOUĂ DIRECȚII, UN SINGUR DIALOG

`input()` aduce date **în** program — este **intrare**. `print()` trimite date **din** program pe ecran — este **ieșire**. **Capcana:** și o linie care *cere* ceva („Introdu vârsta:") este tot un mesaj scris de program, deci tot **ieșire** — programul vorbește primul, apoi ascultă.

A

EXERCIȚIUL A · CLASIFICĂ FLUXUL DE DATE

Cine vorbește acum?

GRILĂ

Mai jos vezi un dialog real dintr-o consolă. Pentru fiecare linie, încercuiește cine a produs-o: **P** = programul a scris-o (ieșire) sau **T** = ai tastat-o tu (intrare).

P = program (ieșire) **T** = tu (intrare)

1	`Introdu numele tău:`	P	T
2	`Maria`	P	T
3	`Introdu vârsta ta:`	P	T
4	`15`	P	T
5	`Salut, Maria!`	P	T
6	`La anul vei avea 16 ani`	P	T
7	`Mai vrei o dată? (da/nu):`	P	T
8	`nu`	P	T

HINT *întreabă-te la fiecare linie: a apărut singură pe ecran (programul a scris-o) sau a apărut fiindcă ai apăsat tu tastele? Liniile care se termină cu* `:` *și cer ceva sunt tot ale programului.*

SECȚIUNEA 2 · TEMA CENTRALĂ

2 Consolă, fereastră, *fișier*

Un program are nevoie să **primească** date și să **livreze** rezultate. Până acum ai folosit un singur drum — consola. Dar nu e singurul. Același program poate comunica în trei feluri, fiecare potrivit pentru altă situație.

INTERFAȚA TEXT

Consola

Programul scrie rânduri de text, tu răspunzi de la tastatură. Simplă, rapidă de scris, bună pentru programe mici. Dezavantaj: trebuie să știi exact ce să scrii și în ce ordine.

FEREASTRA CU BUTOANE

Interfața grafică

Utilizatorul vede o fereastră: apasă butoane, completează casete, citește etichete. Prietenoasă pentru cine nu cunoaște programul. Dezavantaj: mai greu de construit (Tkinter).

MEMORIA DE DURATĂ

Fișierul

Datele vin dintr-un fișier de pe disc, iar rezultatele se scriu tot într-un fișier — rămân și după ce programul se închide. Bun pentru **multe** date deodată.

O IDEE PENTRU TOATĂ LECȚIA

Indiferent de drum, fluxul rămâne același: există o **intrare** și o **ieșire**. Se schimbă doar *cum* arată dialogul, nu *ce* face programul.

A

EXERCIȚIUL A · POTRIVIRE

Cardul potrivit

ASOCIERE

Pentru fiecare situație, încercuiește modalitatea cea mai potrivită: **C** = consolă, **G** = interfață grafică, **F** = fișier. Întreabă-te: *cine* folosește programul și *câte* date sunt la mijloc?

SITUAȚIA	ALEGERE
O aplicație de agendă pentru bunica ta, care nu a mai programat niciodată.	C G F
Un program mic pe care îl scrii repede la ora de informatică, ca să testezi o idee.	C G F
Prelucrarea notelor a 500 de elevi, salvate deja dinainte.	C G F
Un raport care trebuie păstrat ca să fie recitit și săptămâna viitoare.	C G F
Un joc simplu care întreabă „Ce număr alegi?" și așteptă să tastezi.	C G F

HINT *ai multe date sau vrei ca ele să rămână după închidere → fișier. Un utilizator neexperimentat → fereastră. Un program mic, scris repede pentru tine → consolă.*

SECȚIUNEA 3 · INTERFAȚA TEXT

3 Consola, *în detaliu*

Dintre cele trei, consola e singura pe care o poți *scrie* chiar acum. Funcționează cu două instrucțiuni pe care le cunoști: `input()` aduce date în program, `print()` trimite rezultate afară.

> **CE INTRĂ ESTE MEREU TEXT**
>
> Orice tastează utilizatorul, `input()` îi predă programului ca **șir de caractere** (`str`). Dacă ai nevoie de un număr cu care să calculezi, îl convertești: `int(...)` pentru întregi, `float(...)` pentru zecimale. A uita conversia este una dintre cele mai frecvente greșeli.

A EXERCIȚIUL A · PREDICȚIE DE OUTPUT

Anticipează dialogul

GRILĂ

Pentru fiecare cod, execută-l în minte pe intrarea dată și încercuiește ce afișează consola. Atenție la prompter și la conversia text → număr.

Programul 1 — se tastează `2`, apoi `3`.

```
a = input()
b = input()
print(a + b)
```

→ Afișează: 5 23 6

De ce? _____

Programul 2 — se tastează `2`, apoi `3`.

```
a = int(input())
b = int(input())
print(a + b)
```

→ Afișează: 5 23 6

De ce e diferit de Programul 1? _____

Programul 3 — se tastează `15`. Prompterul face parte din ieșire.

```
varsta = int(input("Varsta: "))
print("La anul:", varsta + 1)
```

→ Pe ecran apare, în ordine:

a) `Varsta: 15` apoi `La anul: 16`

b) doar `La anul: 16`

c) `Varsta: 151`

HINT *la Programul 1, `input()` întoarce text, iar `+` între două șiruri le* **lipește** *. La Programul 3, textul din `input("...")` e afișat de program înainte să tastezi — deci apare pe ecran.*

SECȚIUNEA 4 · FEREASTRA CU BUTOANE

4 Interfața *grafică*

Prin consolă, utilizatorul trebuie să știe exact ce să scrie și în ce ordine. Pentru cineva care nu cunoaște programul, e greu. Soluția e **interfața grafică**: o fereastră în care se apasă butoane și se completează casete. Aproape orice fereastră e făcută din patru piese.

> ACEEAȘI LOGICĂ, ALT ÎNVELIȘ · DOAR RECUNOAȘTERE
>
> Fereastra nu schimbă *ce face* programul — schimbă doar *cum arată dialogul*. Codul de mai jos e **doar de citit**: îl vei învăța la lecția despre Tkinter, spre finalul anului.

```
from tkinter import Tk, Label, Entry, Button
fereastra = Tk()
eticheta = Label(fereastra, text="Numele tau:")
eticheta.pack()
caseta = Entry(fereastra)
caseta.pack()
buton = Button(fereastra, text="Saluta")
buton.pack()
fereastra.mainloop()
```

A

EXERCIȚIUL A · ELEMENT ↔ ROL

Identifică piesa

COMPLETARE

Fiecare element din cod are un rol și un **echivalent în consolă**. Completează, pentru fiecare, ce face și cu ce instrucțiune din consolă seamănă (`input`, `print`, `Enter` sau „—" dacă nu are).

ELEMENT FEREASTRĂ	CE FACE	ECHIVALENT CONSOLĂ
`Label`		
`Entry`		
`Button`		
`mainloop()`		

HINT *`Entry` e caseta în care se scrie (ca `input`), `Label` afișează text (ca `print`), `Button` declanșează acțiunea (ca `Enter`). `mainloop()` ține fereastra deschisă — nu are corespondent în consolă.*

B

EXERCIȚIUL B · COMPARAȚIE ARGUMENTATĂ

Consolă sau fereastră?

GRILĂ

Pentru fiecare afirmație, încercuiește: e o trăsătură a **C** (doar consola), a **F** (doar fereastra), sau ceva ce fac **A** (amândouă)?

1. Utilizatorul trebuie să știe exact ce să scrie și în ce ordine. C F A
2. Utilizatorul apasă butoane și completează casete. C F A
3. Are o intrare și o ieșire (un flux de date). C F A
4. Este mai ușor și mai rapid de scris pentru un program mic. C F A
5. Este mai prietenoasă pentru cineva care nu cunoaște programul. C F A

5 SECȚIUNEA 5 · MEMORIA DE DURATĂ

Fișierul — datele care *rămân*

Consola și fereastra presupun o **persoană** care stă în fața programului. Dar ce faci când ai 500 de date de prelucrat? Sau când vrei ca un rezultat să **rămână** după ce închizi programul? Pentru amândouă există a treia modalitate: **fișierul** — un loc pe disc unde datele stau salvate și după ce programul s-a închis.

> **O PARALELĂ CARE AJUTĂ · DOAR RECUNOAȘTERE**
>
> Gândește-te la fișier ca la un **caiet**: consola și fereastra sunt o conversație vorbită (se termină și se uită), fișierul e caietul în care scrii — rămâne, îl reciteşti mâine. În cod: `.read()` aduce date **în** program (ca `input`), `.write()` trimite date **afară** (ca `print`) — dar nu pe ecran, ci într-un fișier care rămâne.

A

EXERCIȚIUL A · COMPLETEAZĂ TABELUL

Cele trei drumuri, comparate

COMPLETARE

Completează căsuțele goale, comparând cele trei modalități pe fiecare rând.

ÎNTREBARE	CONSOLĂ	FEREASTRĂ	FIȘIER
Cine dă datele?	_____	_____	_____
Datele rămân după închidere?	_____	_____	_____
Cum intră datele?	`input()`	casetă	_____
Cum ies rezultatele?	_____	etichetă	_____

HINT *doar la fișier datele rămân după închidere. Instrucțiunile de fișier sunt* `.read()` *(intrare) și* `.write()` *(ieșire).*

B

EXERCIȚIUL B · SURSA DATELOR

De unde vin datele?

GRILĂ

Pentru fiecare program, încercuiește de unde vin **datele de intrare**: **C** = consolă (tastatură), **G** = fereastră (butoane/casete), **F** = fișier.

1. Un program cere numele de la tastatură și te salută. C G F
2. Un program prelucrează notele a 500 de elevi, salvate deja pe disc. C G F
3. Utilizatorul completează o casetă și apasă butonul „Trimite". C G F
4. Un joc întreabă „Câți jucători?" și aștepți să tastezi răspunsul. C G F

6 Detectivul *interfețelor*

SECȚIUNEA 6 · PUNEM TOTUL CAP LA CAP

Ai cunoscut cele trei drumuri. Acum le pui la lucru pe cazuri reale: analizezi programe scrise de alți elevi și decizi dacă au ales **bine** modalitatea de comunicare. A **evalua** o alegere este pasul cel mai matur al gândirii.

A

EXERCIȚIUL A · INVESTIGAȚIE GHIDATĂ

Detectivul de interfețe

COMPLETARE

Trei elevi au făcut alegeri. Pentru fiecare, încercuiește dacă alegerea e **potrivită** și scrie pe scurt **de ce** (sau ce ar fi fost mai bine).

CE A FĂCUT ELEVUL	POTRIVIT?	DE CE / CE AR FI MAI BINE
Radu scrie în consolă un program care adună două numere, pentru o temă la clasă.	DA NU	_____
Ana face o agendă telefonică pentru bunica ei, doar în consolă, cu comenzi text.	DA NU	_____
Vlad calculează media a 300 de note tastându-le una câte una de la tastatură.	DA NU	_____

HINT *o temă mică scrisă pentru tine → consola e potrivită. Un utilizator neexperimentat → mai bine o fereastră. Foarte multe date deja salvate → mai bine un fișier, nu tastarea manuală.*

B

EXERCIȚIUL B · CONSTRUCȚIE PE REGULI

Repară alegerea

ASOCIERE

Scenariu: „Citește notele tuturor elevilor dintr-o clasă (multe, salvate deja) și produce un raport cu media fiecărui elev — raport care trebuie păstrat pentru diriginte, ca să-l deschidă și săptămâna viitoare." Încercuiește alegerea corectă la fiecare decizie.

1 **De unde vin notele?** `tastatură` `fișier` `butoane`

2 **Unde ajunge raportul?** `doar pe ecran` `într-un fișier` `într-o fereastră`

3 **De ce așa?** _____

HINT *„multe date, salvate deja" cere* **intrare din fișier** *; „trebuie păstrat pentru mai târziu" cere* **ieșire în fișier** *— pe ecran s-ar pierde la închidere.*

SECȚIUNEA 7 · COD GHIDAT

7 Programează *dialogul* prin consolă

Până acum ai **citit** despre cele trei modalități. Acum scrii cod adevărat pentru singura pe care o poți rula acum: consola. Programezi dialoguri prin `input` și `print`. Afișează **doar** ce se cere, fără text în plus. Verificarea automată (5 cazuri, prag 4/5) rămâne pe platformă.

1

PROBLEMA 7.1 · DIALOG SIMPLU

Salutul personalizat

SCRIERE COD

Citește un **nume** și o **vârstă** (întreg). Afișează pe primul rând un salut, iar pe al doilea rând câți ani va avea persoana la anul. Ex.: pentru `Ana` și `15` → `Salut, Ana` / `La anul vei avea 16 ani`.

HINT `varsta` *trebuie convertită cu* `int(input())`. *Pentru anul viitor adună 1. Separă cuvintele de valori prin virgulă în* `print`.

SOLUȚIA TA · PYTHON

2

PROBLEMA 7.2 · TREI CATEGORII

Recomandarea după vârstă

SCRIERE COD

Citește **vârsta** (întreg) și afișează secțiunea recomandată: sub 13 ani → `Sectiunea pentru copii`; între 13 și 17 ani → `Sectiunea pentru adolescenti`; de la 18 ani în sus → `Sectiunea pentru adulti`.

HINT *trei categorii → cascadă* `if / elif / else`*. Atenție la limite (13 și 18). Text exact, fără diacritice.*

SOLUȚIA TA · PYTHON

3

PROBLEMA 7.3 · CA DINTR-UN FIȘIER

Citește „fișierul" linie cu linie

SCRIERE COD

Pe primul rând se citește `n` — câte note urmează. Pe următoarele `n` rânduri se citesc note (întregi). Afișează `Note mari:` urmat de câte note sunt **cel puțin 9**. Ex.: `4` apoi `9 7 10 8` → `Note mari: 2`.

HINT *e o punte spre fișiere: primul rând spune câte urmează, apoi* `for` + `range(n)`*. Numeri cu un contor* `mari = mari + 1` *câte au* `nota >= 9`*.*

SOLUȚIA TA · PYTHON

EXERCIȚII DIFERENȚIATE · OPȚIONALE

Următoarele două nu contează la deblocarea evaluării. Unul e pentru consolidare, celălalt e o provocare.

R

REMEDIERE · OPȚIONAL

Salut invers

SCRIERE COD

Citește un **nume** și un **oraș** (ambele text). Afișează pe un singur rând: `Buna, <nume> din <oras>!`
Ex.: `Ana`, `Cluj` → `Buna, Ana din Cluj!`

HINT *fără conversii — totul e text. Atenție la* `!` *de la final, lipit de oraș.*

SOLUȚIA TA · PYTHON

P

PERFORMANȚĂ · OPȚIONAL

Raport corect gramatical

SCRIERE COD

Ca la Problema 1, dar corect: dacă persoana va avea 1 an, scrie „1 an", altfel „N ani". Pe primul rând se citește `n` — câte persoane urmează (ca dintr-un fișier) — apoi pentru fiecare se citesc `nume` și `vârsta`.

HINT *alegi cuvântul cu* `cuvant = "an" if ani == 1 else "ani"`*. Parcurgi persoanele cu* `for _ in range(n)`*.*

SOLUȚIA TA · PYTHON