FIȘĂ DE LUCRU · PYTHON · CLASA A IX-A · FIȘA 10 · LECȚIA PY9-10

Introducere în programarea orientată pe obiecte: folosim clase predefinite

Durată: 50 min Nivel: mate-info · Python

CE VEI ȘTI DUPĂ LECȚIE

- Prezici efectul metodelor și al atributelor pentru obiecte din clasele str, list, Fraction, date și Random.
- Repari erorile tipice: paranteze uitate sau în plus, import greșit, rezultatul unei metode care modifică obiectul.
- Scrii programe scurte care creează obiecte din clase predefinite și le folosesc membrii.

Ce trebuie să știi (o pagină)

Regula	Exemplu	Rezultat
clasa unui obiect	`type(Fraction(1, 2)).__name__`	`'Fraction'`
instanțiere = clasa „chemată” ca o funcție	`datetime.date(2027, 1, 1)`	un obiect `date`
atribut (dată membră): fără paranteze	`d.year`, `f.numerator`	`2027`, numărătorul
metodă: cu paranteze	`d.weekday()`, `s.upper()`	`4`, textul cu majuscule
`str` nu se modifică: păstrezi rezultatul	`s = s.strip()`	
`list` se modifică: metoda întoarce `None`	`a.append(3)`	`a` are un element în plus
`import modul` → `modul.Clasa(...)`	`import random`, `random.Random(5)`	
`from modul import Clasa` → `Clasa(...)`	`from fractions import Fraction`	
fracție dintr-un text	`Fraction("3/4")`, `Fraction("0.1")`	`3/4`, `1/10`
dată dintr-un text ISO	`datetime.date.fromisoformat("2026-10-29")`	
`date - date` dă `timedelta`	`(d2 - d1).days`	numărul de zile
`date + timedelta(days=n)`	`d + datetime.timedelta(days=1)`	ziua următoare
`weekday()`: 0 = luni, ..., 6 = duminică		

Regula	Exemplu	Rezultat
membrii unei clase	`dir(x)`, `help(x.metoda)`	

> **SFAT**
> Înainte să folosești o metodă nouă, pune-ți două întrebări: **întoarce ceva** sau **modifică obiectul**? Primește argumente? Un test de un rând în interpretor îți răspunde imediat.

Nivelul 1 · Prezice

★ citește și prezice

P1. Ce afișează? (Între ghilimele sunt câte două spații la început și la sfârșit.)

```
m = "  Banat 2026  "
print(m.strip().upper(), len(m), len(m.strip()), m.strip().find("2"))
```

P2. Ce afișează?

```
a = [1, 2]
b = a.append(3)
print(a, b, len(a))
```

P3. Ce afișează?

```
from fractions import Fraction
f = Fraction(3, 4) * Fraction(2, 9)
print(f, f.numerator + f.denominator, f < Fraction(1, 5))
```

P4. Ce afișează? (2027 nu este an bisect.)

```
import datetime
d = datetime.date(2027, 3, 1)
e = d - datetime.timedelta(days=1)
print(e, e.day, e.month)
```

P5. Ce afișează prima valoare? Poți prezice, fără să rulezi, a doua valoare?

```
import random
g = random.Random(7)
a = g.randint(1, 100)
h = random.Random(7)
b = h.randint(1, 100)
print(a == b, a)
```

P6. Ce afișează?

```
import datetime
from fractions import Fraction
print(type(Fraction(4, 2)).__name__, Fraction(4, 2) == 2, Fraction(4, 2))
x = datetime.date(2026, 1, 1) - datetime.date(2025, 1, 1)
print(type(x).__name__, x.days)
```

P7. Ce afișează? (`title()` face prima literă a fiecărui cuvânt majusculă.)

```
print("ana are mere".title().replace(" ", ""), "10-03-2026".split("-"))
```

P8. Ce afișează? (`find` întoarce poziția primei apariții sau `-1`.)

```
x = "Python"
print(x.lower().count("p"), x.count("p"), x.find("z"), x.find("t"))
```

Nivelul 2 · Găsește și repară

★★ găsește și repară

P9. Programul se oprește cu `TypeError: 'NoneType' object is not iterable`. De ce? Repară-l.

```
note = [5, 9, 7]
note = note.sort()
print(max(note))
```

P10. Programul dă `ValueError: day is out of range for month`. Ce este greșit? Ce le spui utilizatorilor programului tău despre date de intrare de acest fel?

```
import datetime
d = datetime.date(2026, 9, 31)
```

P11. Eroarea este `TypeError: 'int' object is not callable`. Găsește-o și repar-o.

```
import datetime
d = datetime.date(2026, 9, 15)
print(d.year(), d.weekday())
```

P12. Eroarea este `NameError: name 'Random' is not defined`. Scrie **două** reparații diferite (schimbând câte un singur rând).

```
import random
g = Random(3)
print(g.randint(1, 10))
```

P13. Un elev vrea fracția 1/10 și scrie `print(Fraction(0.1))`. Obține `3602879701896397/36028797018963968`. De ce? Scrie două variante corecte.

P14. *Rânduri amestecate.* Ordonează rândurile ca programul să afișeze câte zile mai sunt de pe 29 octombrie 2026 până pe 3 decembrie 2026. Un rând este în plus: care și de ce nu merge? Ce afișează programul?

```
A)  azi = datetime.date(2026, 10, 29)
B)  import datetime
C)  print((zi - azi).days)
D)  zi = datetime.date(2026, 12, 3)
E)  print(zi.days - azi.days)
```

Nivelul 3 · Scrie cod

★★ scrie cod

Pentru fiecare program, verifică **toate** testele înainte să spui că e gata.

P15. Ziua din calendar. Citește o dată de forma `zz.ll.aaaa` și afișează numele zilei săptămânii, în română (`luni`, `marți`, ..., `duminică`).

intrare	ieșire
`01.01.2027`	`vineri`
`29.02.2028`	`marți`
`01.12.1918`	`duminică`

P16. Câte zile ai trăit? Citește pe două rânduri data nașterii și data de azi, ambele `zz.ll.aaaa`, și afișează numărul de zile dintre ele.

intrare	ieșire
`29.10.2010` și `29.10.2026`	`5844`
`01.01.2026` și `01.01.2026`	`0`
`28.02.2012` și `01.03.2012`	`2`

P17. Suma exactă. Citește de pe un rând fracții scrise `p/q`, separate prin spații, și afișează suma lor, simplificată.

intrare	ieșire
`1/2 1/3 1/6`	`1`
`1/4 1/4`	`1/2`
`2/3`	`2/3`
`3/4 5/6 -1/12`	`3/2`

P18. Parola de unică folosință. Citește un număr `seed`, creează `random.Random(seed)` și construiește o parolă de 8 caractere: la fiecare pas alegi cu `choice` un caracter din `"abcdefghjkmnpqrstuvwxyz23456789"` (fără litere și cifre ușor de confundat) și îl lipești la parolă. Afișează parola. De ce nu e bine ca parolele reale să se obțină dintr-un `seed` cunoscut?

intrare	ieșire
`1`	`ev643cjd`
`2026`	`dmt9tx69`
`42`	`xda2jhhe`

P19. Cenzura. Citești pe două rânduri un text și un cuvânt. Afișezi de câte ori apare cuvântul în text (`count`) și textul în care fiecare apariție e înlocuită cu atâtea `*` câte litere are cuvântul (`replace`). Majusculele contează: `Ana` nu este `ana`.

intrare (două rânduri)	ieșire
`ana are mere si ana are pere` și `ana`	`2 *** are mere si *** are pere`
`Ana are mere` și `ana`	`0 Ana are mere`

intrare (două rânduri)	ieșire
`cer senin` și `luna`	`0 cer senin`

P20. Termenul-limită. Citește o dată `aaaa-ll-zz` și un număr întreg `n` (poate fi și negativ). Afișează data de peste `n` zile, în formatul `zz.ll.aaaa`.

intrare	ieșire
`2026-10-29` și `100`	`06.02.2027`
`2028-02-28` și `1`	`29.02.2028`
`2026-12-31` și `1`	`01.01.2027`
`2026-03-01` și `-1`	`28.02.2026`

Nivelul 4 · Pune mintea la contribuție

★★★ pune mintea la contribuție

P21. Fracția ascunsă. Găsește numerele naturale `p` și `q` cu `p + q = 50` pentru care `print(Fraction(p, q))` afișează `2/3`. Apoi: câte perechi `(p, q)` cu `1 ≤ p, q ≤ 30` dau `Fraction(p, q) == Fraction(2, 3)`? Justifică fără să le numeri pe toate.

P22. Zece de 0,1. Adună de 10 ori `0.1` într-o variabilă `float` și de 10 ori `Fraction(1, 10)` într-o variabilă `Fraction`. Compară fiecare sumă cu `1`. Ce observi și când ai alege `Fraction` în loc de `float`? Dar invers?

P23. Vineri 13. Scrie un program care afișează lunile din anul 2026 în care ziua de 13 cade vinerea. Fă la fel pentru 2027. Câte obiecte `date` creează programul pentru un an?

P24. Verifici cu o clasă predefinită. Scrie funcția `bisect(an)` (an bisect: divizibil cu 4 și nu cu 100, sau divizibil cu 400). Verifică-o automat, pentru toți anii 1900–2100, comparând cu `(datetime.date(an + 1, 1, 1) - datetime.date(an, 1, 1)).days == 366`. Pentru care ani din interval ar greși varianta simplă `an % 4 == 0`?

P25. Mini-proiect: generator de exerciții. Scrie un program care, pentru un `seed` citit, generează 5 exerciții de adunare a două fracții subunitare (numitor între 2 și 9, numărător între 1 și numitor − 1, alese cu același obiect `random.Random(seed)`) și afișează fiecare exercițiu cu rezultatul simplificat. De ce e util ca profesorul și elevul să folosească același `seed`?
