FIȘĂ DE LUCRU · PYTHON · CLASA A IX-A · FIȘA 8 · LECȚIA PY9-08

Subprograme (1): definire, parametri, apel

Durată: 100 min Nivel: mate-info · Python

CE VEI ȘTI DUPĂ LECȚIE

- Prezici rezultatul apelurilor, inclusiv al celor imbricate și al funcțiilor fără return.
- Repari greșelile tipice: print în loc de return, return prea devreme, apel înainte de definiție, antet „ca în C++".
- Scrii funcții cu antetul cerut, le testezi separat și le combini într-un program modular.

Ce trebuie să știi (o pagină)

Noțiune	Exemplu	De reținut
definiție = antet + corp	`def arie(a, b):` ↵ `····return a * b`	corpul rulează abia la apel
parametri	`a`, `b` din antet	primesc valori la fiecare apel
argumente	`arie(20, 15)`, `arie(x, 2 * y)`	se potrivesc **în ordine**, la fel de multe
argumente cu nume	`arie(b=15, a=20)`	ordinea nu mai contează
`return`	`return a * b`	trimite valoarea **și** oprește funcția
mai multe valori	`return q, r` → `q, r = f(…)`	
fără `return`	funcția întoarce `None`	`print` în funcție ≠ `return`
apel în apel	`g(f(3))`	întâi `f(3)`, apoi `g` cu rezultatul
cadru / stivă de apeluri	`a` → `b` → `c`	fiecare apel are variabilele lui; ultimul apelat se termină primul
ordinea în fișier	definițiile înaintea primului apel	altfel apare `NameError`
test pe modul	`assert f(9081) == 18`	testezi funcția înainte de program

Nivelul 1 · Prezice

★ citește și prezice

P1. Numește antetul, corpul, parametrii și argumentele. Ce afișează?

```
def volum(l, L, h):
    return l * L * h

v = volum(2, 3, 4)
print(v, volum(1, 1, v))
```

P2. Ce afișează?

```
def f(x):
    return x + 1
def g(x):
    return 2 * f(x)
print(g(3), f(g(1)), g(f(0)))
```

P3. În ce ordine apar rândurile?

```
def p(x):
    print("p", x)
    return x * x
print(p(2) + p(3))
```

P4. Ce afișează `print(g(10), g(16), g(0))`? Ce returnează `g`, în cuvinte?

```
def g(n):
    for i in range(1, n + 1):
        if i * i > n:
            return i
    return -1
```

P5. Ce afișează?

```
def cat_rest(a, b):
    return a // b, a % b
q, r = cat_rest(17, 5)
print(q, r, cat_rest(9, 3))
```

P6. Ce afișează `r = afis(3)` urmat de `print(r)`, dacă `def afis(n): print(n * "*")`?

P7. Se modifică `a`? Ce afișează?

```
def inc(x):
    x = x + 1
    return x
a = 5
b = inc(a)
print(a, b)
```

P8. Ce afișează `print(putere(2, 5), putere(5, 2), putere(exp=3, baza=2))`, dacă `def putere(baza, exp): return baza ** exp`?

Nivelul 2 · Găsește și repară

★★ găsește și repară

P9. Programul se oprește cu `TypeError: unsupported operand type(s) for +: 'NoneType' and 'NoneType'`. Ce afișează înainte de eroare? Repară funcția.

```
def suma_cifre(n):
    s = 0
    while n > 0:
        s += n % 10
        n //= 10
    print(s)

total = suma_cifre(12) + suma_cifre(34)
print(total)
```

P10. Funcția „verifică” dacă `n` este prim. Ce returnează pentru 9, 2, 15 și 7? Unde este greșeala? Repar-o.

```
def este_prim(n):
    for d in range(2, n):
        if n % d == 0:
            return False
        else:
            return True
```

P11. Programul dă `NameError: name 'arie' is not defined`. De ce? Repară-l fără să schimbi nicio instrucțiune.

```
print(arie(3, 4))
def arie(a, b):
    return a * b
```

P12. *Rânduri amestecate.* Programul citește `a b` și afișează câte numere prime sunt în intervalul `[a, b]` (un număr este prim dacă are exact 2 divizori). Un rând este în plus. Indentarea o stabilești tu.

```
A)  def nr_divizori(n):
B)  c = 0
C)  for d in range(1, n + 1):
D)  if n % d == 0:
E)  c += 1
F)  return c
G)  a, b = map(int, input().split())
H)  k = 0
I)  for x in range(a, b + 1):
J)  if nr_divizori(x) == 2:
K)  k += 1
L)  print(k)
M)  print(c)
```

Ce afișează pentru `1 20`? Dar pentru `90 96`?

P13. *Traducere din C++.* Ce rânduri dau eroare în Python și care sunt doar „neobișnuite”, dar merg?

```
int maxim(a, b):
    if (a > b)
        return a;
    return b;
```

P14. Un elev vrea viteza medie pentru 120 km parcurși în 2 ore. Scrie `print(viteza(2, 120))` și obține `0.016666666666666666`. Apoi încearcă `viteza(120)` și primește o eroare. Explică ambele situații și scrie un apel care nu depinde de ordinea parametrilor.

```
def viteza(distanta, timp):
    return distanta / timp
```

Nivelul 3 · Scrie cod

★★ scrie cod

Respectă **exact** antetul cerut. Testează fiecare funcție separat, apoi programul.

P15. Coșul de cumpărături. `cost_total(pret, cantitate, reducere)` returnează costul după aplicarea reducerii (în procente). Afișează cu 2 zecimale.

apel	rezultat
`cost_total(12.5, 4, 10)`	`45.00`
`cost_total(100, 1, 0)`	`100.00`
`cost_total(3, 0, 50)`	`0.00`
`cost_total(19.99, 3, 25)`	`44.98`

P16. Temperatura maximă. Scrie `maxim(a, b)`, apoi un program care citește temperaturile a 7 zile, de pe un rând, și afișează maximul. Folosește **numai** `maxim` pentru comparații (fără `max`).

intrare	ieșire
`3 5 -1 0 8 2 4`	`8`
`-5 -3 -9 -4 -7 -6 -8`	`-3`
`2 2 2 2 2 2 2`	`2`

P17. Palindroame. Scrie `oglindit(n)` și `este_palindrom(n)` (care o folosește pe prima). Afișează palindroamele din `[a, b]`, pe un rând.

intrare	ieșire
`10 50`	`11 22 33 44`
`1 9`	`1 2 3 4 5 6 7 8 9`
`120 135`	`121 131`

P18. Fracții, a doua oară. Scrie `cmmdc(a, b)`, `simplifica(x, y)` (returnează **două** valori, cu numitorul pozitiv) și `aduna(a, b, c, d)` (returnează fracția `a/b + c/d`, simplificată). Compară lungimea codului cu varianta fără funcții de la fișa 6.

apel	rezultat
`aduna(3, 4, 5, 6)`	`19 12`
`aduna(1, 2, 1, 2)`	`1 1`

apel	rezultat
`aduna(1, -2, 1, 3)`	`-1 6`
`aduna(1, 2, -1, 2)`	`0 1`

P19. Numere prietene. `suma_divizori(n)` returnează suma divizorilor lui `n` **mai mici decât** `n`. Două numere diferite `a`, `b` sunt prietene dacă fiecare este suma divizorilor celuilalt. Scrie `sunt_prietene(a, b)`.

intrare	ieșire
`220 284`	`DA`
`1184 1210`	`DA`
`10 20`	`NU`
`6 6`	`NU`

P20. Cronometrul. `hms(s)` returnează orele, minutele și secundele dintr-un număr de secunde. Programul afișează `hh:mm:ss`.

intrare	ieșire
`3725`	`01:02:05`
`59`	`00:00:59`
`86399`	`23:59:59`

Nivelul 4 · Pune mintea la contribuție

★★★ pune mintea la contribuție

P21. Cu sau fără subprograme? Programul numără primele din `1…100` apelând `este_prim(x)` (varianta cu `d * d <= n` din lecție) pentru fiecare `x`. (a) De câte ori este apelată funcția? (b) De câte ori se execută în total bucla `while` din ea? (c) Programul ar fi mai rapid fără funcție, cu bucla scrisă direct în `for`? Argumentează ce câștigi și ce (nu) pierzi.

P22. Problemă inversă. Pentru ce valori `1 ≤ b ≤ 30` returnează `f(24, b)` valoarea `6`?

```
def f(a, b):
    while b:
        a, b = b, a % b
    return a
```

P23. Stiva de apeluri. Ce afișează `print(a(3))`? Desenează stiva de cadre în momentul în care se execută `c` **prima** dată. Câte cadre sunt pe stivă (cu tot cu programul principal)?

```
def c(x): return x * 10
def b(x): return c(x + 1) - 1
def a(x): return b(2 * x) + c(x)
```

P24. Proiectează întâi, scrie apoi. Un program citește notele unui elev, de pe un rând, afișează media cu 2 zecimale și `promovat` (media ≥ 5) sau `corigent`. (a) Împarte problema în **trei** funcții:

pentru fiecare scrie antetul, ce primește și ce returnează. (b) Implementează și testează.

intrare	ieșire
`7 8 5`	`6.67 promovat`
`4 5 4`	`4.33 corigent`
`4 6 5`	`5.00 promovat`
`10`	`10.00 promovat`

P25. Mini-proiect: calculatorul de conversii. Scrie funcțiile `km_in_mile(x)` (1 milă = 1,609344 km), `celsius_in_f(c)` (`F = C · 9/5 + 32`) și `lei_in_euro(suma, curs)`. Apoi scrie un program principal care citește o opțiune (`1`, `2` sau `3`) și valorile necesare, apelează funcția potrivită și afișează rezultatul. La o opțiune greșită afișează `opțiune invalidă`. Prezintă proiectul colegilor: ce face fiecare funcție?

intrare	ieșire
`1 100`	`62.14`
`2 37`	`98.6`
`3 500 4.97`	`100.60`
`7 1`	`opțiune invalidă`

Probleme de lucru în clasă (pbinfo)

Le rezolvăm împreună în ședință, în ordinea de mai jos.

- [ ] **#368** **nrpal** — ușoară
  https://www.pbinfo.ro/probleme/368/nrpal
- [ ] **#64** **numarperfect** — ușoară
  https://www.pbinfo.ro/probleme/64/numarperfect

Temă pentru acasă (pbinfo)

Trimite soluțiile pe pbinfo și notează punctajul obținut. Bifează ce ai terminat.

- [ ] **#1574** **prietene** — ușoară
  https://www.pbinfo.ro/probleme/1574/prietene
- [ ] **#375** **urmatorul-prim** — ușoară
  https://www.pbinfo.ro/probleme/375/urmatorul-prim