FIȘĂ DE LUCRU · PYTHON · CLASA A IX-A · FIȘA I4 (INTENSIV) · LECȚIA PY9-I4

Fișiere text în C++: ifstream și ofstream

Durată: 100 min Nivel: intensiv · Python + C++

CE VEI ȘTI DUPĂ LECȚIE

- Prezici ce citește >> dintr-un fișier și ce ajunge într-un fișier scris cu <<.
- Repari programele C++ cu fișiere: bucla cu eof(), close() uitat, deschideri greșite, traduceri greșite din Python.
- Scrii programe C++ care citesc n valori sau până la sfârșitul fișierului și tratezi fișierul lipsă sau gol.
- Compari comportamentul și viteza celor două limbaje la lucrul cu fișiere.

Ce trebuie să știi (o pagină)

Ce vrei	Python	C++
biblioteca		`#include <fstream>` (`<iomanip>` pentru `setprecision`)
deschizi pentru citire	`with open("a.in") as f:`	`ifstream f("a.in");`
deschizi pentru scriere / adăugare	`open("a.out", "w")` / `open("a.out", "a")`	`ofstream g("a.out");` / `ofstream g("a.out", ios::app);`
citești un număr	`int(f.readline())` (un rând)	`f >> n;` (sare peste spații și Enter-uri)
citești până la capăt	`for x in f.read().split():`	`while (f >> x) { … }`
citești un cuvânt / un rând	`split()` / `f.readline()`	`f >> s;` / `getline(f, s);` (`s` de tip `string`)
scrii	`print(x, y, file=g)`	`g << x << ' ' << y << '\n';` (nimic automat)
două zecimale	`f"{m:.2f}"`	`g << fixed << setprecision(2) << m;`
fișierul lipsește	`FileNotFoundError`	nicio eroare; verifici cu `if (!f) { … }`
date greșite (`x` în loc de număr)	`ValueError`	citirea eșuează în tăcere, bucla se oprește
închizi	automat la ieșirea din `with`	`f.close();` sau automat la sfârșitul lui `main`

ATENȚIE — GREȘELI FRECVENTE

`while (!f.eof())` este greșit: `eof()` devine adevărat abia **după** o citire ratată, deci bucla mai face un pas cu valoarea veche. Folosește `while (f >> x)`.

În enunțuri, `↵` marchează sfârșitul unui rând. În toate programele C++ se presupun `#include <fstream>`, `#include <iostream>`, `#include <string>` și `using namespace std;`, dacă nu se spune altfel.

Nivelul 1 · Prezice

★ citește și prezice

P1. `p1.in` conține `4↵2 9 4 1 7 3↵`. Ce afișează? Ce se întâmplă cu valorile `7` și `3`?

```
ifstream f("p1.in");
int n, x, s = 0;
f >> n;
for (int i = 0; i < n; i++) { f >> x; s += x; }
cout << s;
```

P2. `p2.in` conține `12↵↵ 5 8↵`. Ce valori primesc `a`, `b`, `c` după `f >> a >> b >> c;` și cât este `a - b * c`?

P3. `p3.in` conține `3 4 10 7 8↵`. Ce conține `p3.out` după rulare?

```
ifstream f("p3.in");
ofstream g("p3.out");
int x;
while (f >> x)
    if (x % 2 == 0) g << x;
```

P4. `p4.in` conține `Ana 9 Bogdan 10↵Cris 8↵`. Ce afișează?

```
ifstream f("p4.in");
string nume, best;
int nota, mx = 0;
while (f >> nume >> nota)
    if (nota > mx) { mx = nota; best = nume; }
cout << best << ' ' << mx;
```

P5. Programul este rulat de trei ori la rând, pornind fără fișierul `rulari.txt`. Ce afișează la a treia rulare? Dar dacă scoți `, ios::app`?

```
ofstream g("rulari.txt", ios::app);
g << 1 << '\n';
g.close();
ifstream f("rulari.txt");
int x, c = 0;
while (f >> x) c++;
cout << "Rularea nr. " << c;
```

P6. `p6.in` conține `10 20↵30↵`. Ce afișează fiecare program?

```python
with open("p6.in") as f:
    print(len(f.readline().split()), len(f.readline().split()))
```

```cpp
ifstream f("p6.in");
int x, y, z;
f >> x; f >> y; f >> z;
cout << x + y + z;
```

P7. `date.txt` conține `5 6 7↵`. Ce conține `date.txt` după rularea programului? De ce?

```cpp
ifstream f("date.txt");
ofstream g("date.txt");
int x, s = 0;
while (f >> x) s += x;
g << s << '\n';
```

P8. `p8.in` conține `3.7 2↵`. Ce afișează `int a = -1, b = -1; f >> a >> b; cout << a << ' ' << b;`? (Din C++11, o citire de număr eșuată pune 0 în variabilă.) Ce s-ar fi întâmplat în Python cu `int("3.7")`?

Nivelul 2 · Găsește și repară

★★ găsește și repară

P9. `cheltuieli.in` conține `45 120 30 15↵`. Programul afișează `5 cheltuieli, total 225 lei`. Explică cele două valori greșite și repară bucla.

```cpp
ifstream f("cheltuieli.in");
int x, total = 0, nr = 0;
while (!f.eof()) {
    f >> x;
    total += x;
    nr++;
}
cout << nr << " cheltuieli, total " << total << " lei\n";
```

P10. *Programul lacunar.* `numar.in` conține un număr natural nenul `n`. Programul scrie în `numar.out`, pe trei rânduri: n^2, rădăcina pătrată a lui `n` cu două zecimale și divizorii lui `n`. Completează spațiile. Ce conține `numar.out` pentru `n = 36`? (Varianta Python este P12 din fișa fp9-13.)

```
#include <fstream>
#include ______        // pentru sqrt
#include ______        // pentru setprecision
using namespace std;
int main() {
    ______ f("numar.in");
    ______ g("numar.out");
    int n;
    f ___ n;
    g ___ n * n << '\n';
    g << ______ << setprecision(2) << sqrt(n) << '\n';
    for (int d = 1; d <= n; d++)
        if (______)
            g << d << ' ';
    g << '\n';
    f.close();
    ______;
    return 0;
}
```

P11. `p11.in` conține `4↵7 8 10 6↵`. Fișierul `p11.out` trebuie să fie `Media: 7.75↵Maxim: 10↵`, dar este `Media: 7Maxim: 10`. Găsește **trei** greșeli.

```
#include <fstream>
using namespace std;
int main() {
    ifstream f("p11.in");
    ofstream g("p11.out");
    int n, x, s = 0, mx = 0;
    f >> n;
    for (int i = 0; i < n; i++) { f >> x; s += x; if (x > mx) mx = x; }
    g << "Media: " << s / n;
    g << "Maxim: " << mx;
    return 0;
}
```

P12. *Rânduri amestecate.* Ordonează rândurile din corpul lui `main` astfel încât programul să scrie în `cheltuieli.out` totalul și cea mai mare cheltuială din `cheltuieli.in` (oricâte valori). Un rând este în plus. Ce scrie pentru fișierul din P9?

```
A)  total += x;
B)  ofstream g("cheltuieli.out");
C)  while (f >> x) {
D)  g << total << ' ' << mx << '\n';
E)  ifstream f("cheltuieli.in");
F)  if (x > mx) mx = x;
G)  int x, total = 0, mx = 0;
H)  while (!f.eof()) {
I)  }
```

P13. Programul calculează media valorilor din fișier. Ce se întâmplă dacă fișierul este gol? Dar dacă lipsește? Repară-l astfel încât în ambele cazuri să afișeze un mesaj potrivit (mesaje diferite).

```cpp
ifstream f("date.in");
int x, s = 0, c = 0;
while (f >> x) { s += x; c++; }
cout << s / c << '\n';
```

P14. *Traducere greșită.* Programul Python adaugă un rând în jurnalul unui server. Traducerea în C++ are **trei** greșeli: una de compilare și două care strică fișierul. Găsește-le.

```python
with open("log.txt", "a") as g:
    g.write("10:42 404\n")
```

```cpp
#include <iostream>
using namespace std;
int main() {
    ofstream g("log.txt");
    g << "10:42" << 404;
    return 0;
}
```

Nivelul 3 · Scrie cod

★★ scrie cod

Scrie programe C++ complete. Creează fișierele de intrare exact ca în tabel și verifică **toate** testele.

P15. Cheltuielile lunii. `cheltuieli.in` conține `n`, apoi `n` sume în lei (numere reale). Scrie în `cheltuieli.out` totalul, cu două zecimale, și, pe rândul următor, câte cheltuieli depășesc 100 de lei.

`cheltuieli.in`	`cheltuieli.out`
`4↵45.5 120 30.25 100↵`	`295.75↵1↵`
`1↵99.99↵`	`99.99↵0↵`
`3↵150 200.5 101↵`	`451.50↵3↵`

P16. Înghețul. `temp.in` conține temperaturile zilnice (numere întregi, oricâte, puse oricum pe rânduri). Scrie în `temp.out` cel mai mare număr de zile **consecutive** cu temperatură negativă.

`temp.in`	`temp.out`
`3 -1 -4 2 -2 -5 -1 0 -3↵`	`3↵`
`5 7 2↵`	`0↵`
(gol)	`0↵`
`-1↵-2↵-3` (fără `↵` la final)	`3↵`

P17. Jurnalul serverului. Fiecare rând din `server.log` are forma `ora cod`, de exemplu `08:01 404`. Scrie în `server.out`, pe un rând, câte coduri sunt de tip 2xx (200…299), 4xx și 5xx. Celelalte coduri se ignoră.

`server.log`	`server.out`
`08:00 200↵08:01 404↵08:05 500↵08:07 201↵08:09 503↵`	`2 1 2↵`

server.log	server.out
(gol)	`0 0 0↵`
`12:00 301↵12:01 404`	`0 1 0↵`

P18. Absențele. `absente.in` conține `n`, apoi `n` rânduri `nume motivate nemotivate`. Scrie în `absente.out` elevii cu cel puțin 10 absențe nemotivate (`nume nemotivate`, în ordinea din fișier), iar pe ultimul rând câți sunt.

absente.in	absente.out
`4↵Ana 3 0↵Bogdan 2 12↵Cris 0 10↵Dan 5 9↵`	`Bogdan 12↵Cris 10↵2↵`
`1↵Ela 20 0↵`	`0↵`
`2↵Ion 0 15↵Mara 1 10`	`Ion 15↵Mara 10↵2↵`

P19. Un program care nu se lasă păcălit. `date.in` conține numere întregi. Scrie în `date.out` minimul și maximul, separate prin spațiu. Dacă fișierul lipsește, scrie `EROARE: fisier lipsa`; dacă e gol, scrie `GOL`. Nu folosi valori „santinelă” de tipul `mn = 1000`: numerele pot fi oricât de mari.

date.in	date.out
(nu există)	`EROARE: fisier lipsa↵`
(gol)	`GOL↵`
`7 -3 12 0↵`	`-3 12↵`
`5`	`5 5↵`

P20. Separatorul. Citește numerele întregi din `numere.in` (oricâte) și scrie-le pe cele pare în `pare.out` și pe cele impare în `impare.out`, câte unul pe rând, în ordinea din fișier. Pe ecran afișează câte pare și câte impare au fost.

numere.in	pare.out	impare.out	ecran
`3 8 5 12 0 7↵`	`8↵12↵0↵`	`3↵5↵7↵`	`3 3`
(gol)	(gol)	(gol)	`0 0`

Nivelul 4 · Pune mintea la contribuție

★★★ pune mintea la contribuție

P21. Fișier sau tastatură? Generează din Python un fișier `mare.in` cu un milion de numere naturale aleatoare (`random.randint(1, 10**6)`). Calculează suma lor în patru feluri și măsoară timpul (de exemplu cu `time` în terminal): C++ cu `ifstream` și `while (f >> x)`; C++ cu `cin` și redirectare (`./prog < mare.in`); Python cu `f.read().split()`; Python cu `input()` și redirectare. Ce variabilă trebuie să folosești pentru sumă în C++ și de ce? Cât ar dura dacă datele ar fi tastate chiar de om? Ce schimbă rândul `ios::sync_with_stdio(false);` pus la începutul variantei cu `cin`?

P22. De ce se oprește? `p22.in` conține `1 2 3 4 5↵`. Ce afișează `while (f >> a >> b) cout << a + b << ' ';`? Explică ce se întâmplă cu `5`. Modifică programul astfel încât să anunțe valoarea rămasă fără pereche.

P23. Când nu se vede bug-ul. Bucla `while (!f.eof()) { f >> x; … }` prelucrează de două ori ultima valoare dacă fișierul se termină cu `↵`. Pentru fiecare cerință, spune dacă rezultatul iese totuși corect: (a) maximul; (b) suma; (c) numărul de valori; (d) „există o valoare egală cu 7?"; (e) ultima valoare. Verifică pe `3 8 5↵`. Ce test ai pune ca să prinzi bug-ul la o problemă de tip maxim?

P24. Mini-proiect: zarurile. Un program Python generează `zaruri.txt`:

```
import random
random.seed(2026)
with open("zaruri.txt", "w") as g:
    for i in range(600):
        g.write(str(random.randint(1, 6)) + "\n")
```

Scrie un program C++ care citește `zaruri.txt` (dacă lipsește, afișează un mesaj), apoi afișează de câte ori a apărut fiecare față (cu un tablou de frecvențe, vezi lecția py9-i2), media aruncărilor cu trei zecimale și cea mai lungă serie de aruncări consecutive egale. Verifică frecvențele și media cu un program Python de 3 rânduri. Seamănă rezultatul cu ce te aștepți de la un zar corect?

P25. Numere cu virgulă între ele. `csv.in` conține un singur rând de forma `4,7,9,12`. De ce `while (f >> x) s += x;` dă `4`? Scrie varianta C++ corectă folosind o variabilă `char virgula` și varianta Python cu `split(",")`. Teste: `4,7,9,12 → 32`, `5 → 5`, `10,-3 → 7`, `4, 7 ,9 → 20`.

Probleme de lucru în clasă (pbinfo)

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

- [ ] **#18** numarul-de-divizori — ușoară
  https://www.pbinfo.ro/probleme/18/numarul-de-divizori
- [ ] **#2275** minimdoua — ușoară
  https://www.pbinfo.ro/probleme/2275/minimdoua
- [ ] **#1881** platou4 — ușoară
  https://www.pbinfo.ro/probleme/1881/platou4

Temă pentru acasă (pbinfo)

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

- [ ] **#3391** sirmaxim — ușoară
  https://www.pbinfo.ro/probleme/3391/sirmaxim
- [ ] **#2269** maximpar — ușoară
  https://www.pbinfo.ro/probleme/2269/maximpar
- [ ] **#8** maxim3 — ușoară
  https://www.pbinfo.ro/probleme/8/maxim3