Programare Orientată Obiect

Laborator 9

Despre ce vorbim azi

- Lucrul cu obiectele din clasa *iostream*
- Lucrul cu fișiere

cin și cout

- Rol de citire de la consola, respectiv afișare a datelor pe ecran
- Înlocuiesc funcțiile de programare procedurala printf și scanf
- Sunt obiecte de tip istream, respectiv ostream
- Săgețile indică ordinea de curgere a informației
- istream și ostream fiind clase, includ în constructori operații de deschidere a fișierelor și în destructori operații de închidere a lor, astfel încât programatorul să fie scutit de acest lucru

Metode specifice clasei istream

- cin nu citește șiruri de caractere cu spații (asemănător cu scanf cu identificatorul de format %s)
- Pentru acest lucru putem folosi funcția *gets()* din namespace-ul stdio sau metoda *getline()* a clasei *istream*
- Ex: Pentru citirea unui nume procedăm în felul următor

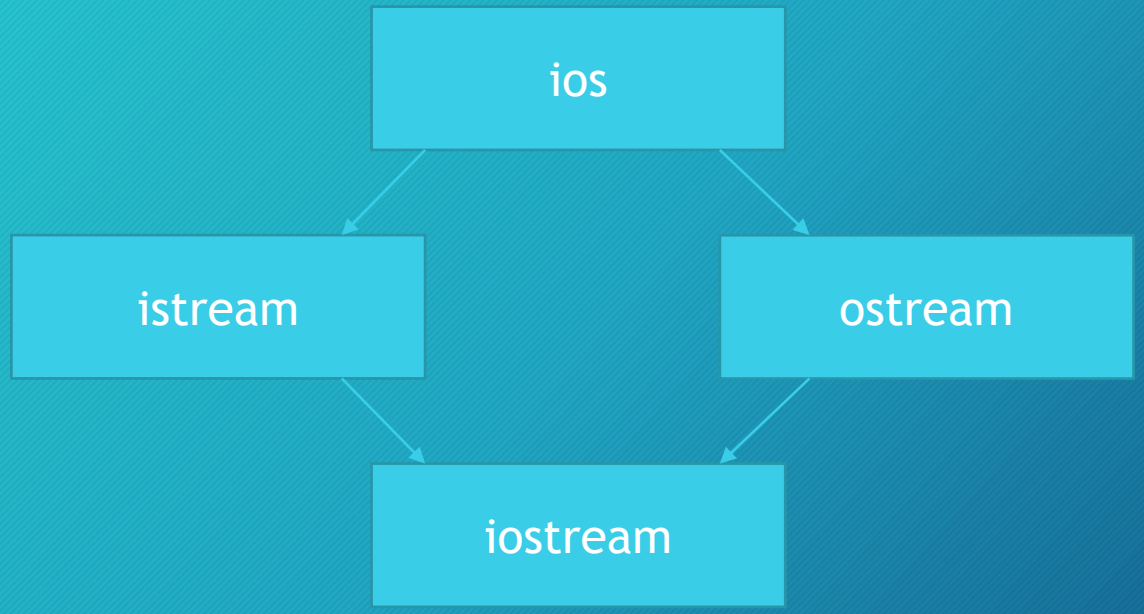
```
char nume[30];
cin.getline(nume, 30);
```

Metode specifice clasei ostream

- O metodă specifică este *write()* ce permite scrierea unui șir de lungime fixă (Ex: cout.write(nume, strlen(nume));)
- O alta este *put()* ce permite scrierea unui singur caracter
- Similar există metoda *get()* în clasa istream ce permite citirea unui singur caracter
- Exemplu:

```
char a;
a = cin.get();
cout.put(a);
```

Ierarhia claselor implicate în lucrul cu ecranul



Formatarea datelor

- Clasa ios dispune de diverse metode (get-eri și set-eri în general) ce ne permit diverse formatări la afișarea datelor
- Exemple: ios::fill, ios::width, ios::precision
- Folosire:

```
int x=123;
cout.fill('#'); cout.width(5);
cout<<x;
//cout<<setfill('#')<<setw(10)<<x<<endl;
```

Efect: ##123

Manipulatori

- Funcții speciale care se plasează în lanțul de operatori << sau >>
- Exemple: dec, hex, oct, setprecision(int), endl, ends, ws, flush, setbase(int), setfill(int), setw(int), setiosflags(long), resetiosflags(long)
- Pentru folosirea manipulatorilo cu argumente avem nevoie de fișierul *iomanip.h*
- Constante ce pot fi trimise ca parametri funcțiilor setiosflags și resetiosflags: ios::left, ios::right, ios::internal, ios::dec, ios::hex, ios::oct, ios::showpos, ios::showbase, ios::scientific, ios::showpoint, ios::uppercase

Lucrul cu fișiere

- Pentru lucrul cu fișiere în C++ putem utiliza clasele ifstream, ofstrem sau fstream situate în fișierul header fstream.h
- Deschidere fișierului

```
ifstream(“fisier.dat”); //prin apelul constructorului
ofstream f;
f.open(“fisier.dat”); //prin apelul unei metode specifice
```

Modalități de deschidere

- Folosind constante specifice
- Constantele se pot aplica în cascadă folosind operatorul logic de sau pe biți | (numit și pipe)
- Constante specifice: ios::in, ios::out, ios::ate, ios::app, ios::trunc, ios::nocreate, ios::noreplace, ios::binary

Modalități de scriere

- În mod text folosind operatorii << și >>
- În mod binar folosind metodele *get* și *put*
- În mod binar folosind metodele *read* și *write*

Deplasarea în fișiere binare

- Se face folosind funcțiile *seekg* (pentru fișiere de intrare) și *seekp* (pentru fișiere de ieșire)
- Cele două metode ai câte două supraîncărcări: una cu un singur parametru în care se specifică numarul de octeți de la începutul fișierului cu care se face deplasarea și una cu doi parametri (numărul de octeți și poziția de referință)
- Poziția de referință poate fi: ios::beg, ios::cur, ios::end
- Aflarea poziției curente din fișier se face cu ajutorul metodelor *tellp*, respectiv *tellg*

Închiderea fișierelor binare

- Prin apelul metodei *close()*