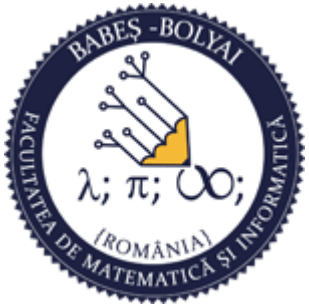
UNIVERSITATEA BABEŞ-BOLYAI
Facultatea de Matematică şi Informatică



Programare orientată obiect

Curs 12

Laura Dioşan

POO

- Operaţii de intrare/ieşire (IO)
  - Fişiere de I/O
  - Biblioteci standard
    - Biblioteca Standard de I/O
    - Biblioteca Standard pentru *stream*-uri I/O

Intrări şi ieşiri

- Facilităţile I/O nu fac parte din limbajul C propriu-zis
- Toate operaţiile de I/o trebuie realizate prin intermediul unor funcţii speciale, precum
  - *getchar(),*
  - *scanf(),*
  - *printf()*, etc.
- Ele nu fac parte din limbajul propriu-zis ➔ pot să difere de la o maşină la alta
- Standardul ANSI

Fişiere IO

- Tipologie
  - Fişiere text
    - Informaţia este reţinută ca vectori de caractere
    - Sfârşitul de fişier → există fizic în fişier (***fgetc()*** returns EOF (-1))
  - Fişiere binare
    - Informaţia este reţinută în formatul original (intern):
      - *int* → 2 octeşi
      - *float* → 4 octeţi, etc.
    - Sfârşitul de fişier → nu există fizic în fişier (***fgetc()*** îl va genera)

- Procesarea fişierelor la 2 nivele:
  - Nivel inferior
    - Acces direct la SO
  - Nivel superior
    - Utilizarea de funcţii de I/O speciale

Procesarea la nivel inferior

- Deschiderea unui fişier
- Închiderea fişierului
- Citirea din fişier
- Scrierea în fişier
- Accesul în fişier

Procesarea la nivel inferior

- Deschiderea unui fişier

  int open(const char* path, int access)

  - acces
    - O_RDONLY
    - O_WRONLY
    - O_CREATE
    - O_RDWR
    - O_APPEND
    - O_BINARY
    - O_TEXT
  - return
    - descriptor de fişier
    - -1 – error
  - headers:
    - <io.h>, <fcntl.h>

Procesarea la nivel inferior

- Închiderea unui fişier

```
int close(int handle)
```

  - handle
    - Descriptor de fişier
  - return
    - 0 – închidere cu succes
    - -1 – eroare

Procesarea la nivel inferior

- Citirea dintr-un fişier

int read(int handle, void* buffer, unsigned length)

  - handle
    - 0 - stdin
    - 1 – stdout
    - 2 – stderr
    - 3 – stdprn
    - 4 - stdaux
  - buffer
    - Zona de memorie în care se citeşte
  - length
    - Numărul maxim de octeţi citiţi
  - return
    - # de octeţi citiţi
    - 0 – la sfârşit de fişier
    - -1 – eroare

Procesarea la nivel inferior

- Scrierea în fişier

  int write(int handle, void* buffer, unsigned length)

  - handle
    - 0 - stdin
    - 1 – stdout
    - 2 – stderr
    - 3 – stdprn
    - 4 - stdaux
  - buffer
    - Zona de memorie din care se preia informaţia care trebuie afişată
  - length
    - # maxim de octeţi scrişi
  - return
    - # de octeţi scrişi
    - -1 – eroare

Procesarea la nivel inferior

- Accesul în fişier

  *long **lseek**(int handle, long offset, int origin)*

  - handle
    - Descriptor de fişier
  - offset
    - Numărul (cu semn) de octeţi după care se face poziţionarea în fişier (relativ la origine)
  - origin
    - Poziţia referinţă
      - SEEK_SET → începutul fişierului
      - SEEK_CUR → poziţia curentă
      - SEEK_END → sfârşitul fişierului
  - return
    - Numărul de octeţi după care se face poziţionarea în fişier (relativ la poziţia originală)
    - -1 – eroare

Procesarea la nivel superior

- Deschiderea unui fişier
- Închiderea fişierului
- Procesarea bazată pe caractere
- Procesarea bazată pe şiruri de caractere
- Procesarea bazată pe fluxuri (*stream*-uri)
- Citirea din fişier
- Scrierea în fişier

- ➔ Biblioteca Standard → stil C
- ➔ Biblioteca pentru fluxuri de I/O - Standard Input / Output Streams Library → stil C++

Biblioteca standard

- O bibliotecă NE-orietată obiect care oferă functionaţităţi pentru operaţiile de I/O
- Operaţii de I/O standard (consolă, respectiv tastatură)
- Operaţii de I/O cu fişiere

Operații standard de I/O

- Operaţii de intrare
  - *int* ***getchar****()*
    - Preia un singur caracter de la consolă
    - Întoarce caracterul de intrare sau EOF
  - *int* ***getc****(FILE*)*
    - Preia un singur caracter de la consolă sau dintr-un fişier
    - Întoarce caracterul de intrare sau EOF
  - *char** ***gets****(char*)*
    - Citeşte un string de la consolă
    - Returnează un pointer către stringul citit sau NULL (în caz de erorare)
- Operaţii de ieşire
  - *int* ***putchar****(int)*
    - Scrie un caracter pe consolă
    - Returnează caracterul scris în caz de succes sau EOF în caz de eroare
  - *int* ***putc****(int, FILE*)*
    - Scrie un caracter pe consolă sau într-un fişier
    - Returnează caracterul scris în caz de succes sau EOF în caz de eroare
  - *int* ***puts****(char*)*
    - Scrie un string pe consolă
    - Returnează un număr ne-negativ în caz de succes sau EOF în caz de eroare

```
char c;
while ((c = getchar()) != EOF){
      putchar(c);
}
```

```
char c;
while ((c = getc(stdin)) != EOF){
        putc(c, stdout);
}
```

```
char s[20];
while (gets(s) != NULL)
                puts(s);
```

Operații standard de I/O

- Operaţii de intrare formatate
  - *int **scanf**(char* format, arg1, arg2, …)*
    - Citeşte un string formatat de la consolă
    - Returnează numărul de elemente citite şi care s-au potrivit cu formatul sau EOF
  - *int **sscanf**(char* string, char* format, arg1, arg2, …)*
    - Citeşte un string formatat dintr-un alt string
- Formatted output operations
  - *int **printf**(char* format, arg1, arg2, …)*
    - Tipăreşte un string formatat pe consolă
    - Returnează numărul de caractere tipărite
  - *int **sprintf**(char* string, char* format, arg1, arg2, …)*
    - Tipăreşte un string formatat în alt string
- Format:
  - %d → valori intregi în baza 10
  - %o → valori intregi în baza 8
  - %x → valori intregi în baza 16
  - %x → valori intregi în baza 16 (upper-case)
  - %f → valori reale
  - %c → caractere
  - %s → stringuri

Exemplu

- A se consulta exemplul din directorul *11/streams*, funcţiile:
  - *standardIO()*
  - *getPut()*

Operații de I/O cu fișiere

- Deschiderea unui fișier
- Citirea din fișier
- Scrierea în fișiere
- Închiderea unui fișier

Deschiderea unui fişier

- Declararea unei variabile de tip fişier
  - *FILE* fileVariableş*
- Deschiderea unui fişiere
  - *FILE ****fopen****(char* fileName, char* access);*
- Access:
  - "r" – deschidere pentru citire
  - "w" – deschidere pentru scriere
  - "a" – deschidere pentru adăugare
  - "r+" – deschidere pentru citire/scriere
  - "rb" – deschidere pentru citire în format binar
  - "wb" – deschidere pentru scriere în format binar
  - "r+b" – deschidere pentru citire/scriere în format binar

Închiderea unui fişiere

- *int **fclose**(FILE *pf);*
  - returnează 0 – succes
  - returnează 1 – erori

Operații de I/O cu fișiere

Tip fişier	Conversie	Unitatea transferată	Funcţii
text	Fără	Caracter	`fgetc()` ⇔ `getc()` `fputc()` ⇔ `putc()`
		Linie	`fgets()` ⇔ `gets()` `fputs()` ⇔ `puts()`
	Cu	Linii	`fscanf()` `fprintf()`
binar	fără	Articol	`fread()` `fwrite()`

Operații de I/O cu fișiere

- *int **fscanf**(FILE* stream, char* format, arg1 , arg2, …)*
  - Citeşte dintr-un fişier un string, caracter sau o dată numerică formatat(ă)
  - Returnează numărul de elemente corect citite
- *int **fprintf**(FILE* stream, char* format, arg1, arg2, …)*
  - Afiseză un string formatat într-un fişier
  - Returnează numărul de elemente afişate sau 0 (eroare)
- *size_t **fread**(void *p, size_t elemSize, size_t noOfElem, FILE *stream)*
  - Citeşte dintr-un fişier binar
  - Returnează numărul de elemente corect citite
- *size_t **fwrite**(void* p, size_t elemSize, size_t noOfElem, FILE* stream)*
  - Scrie într-un fişier binar
  - Returnează numărul de elemente scrise sau 0 (eroare)

Exemple

- A se consulta exemplul din directorul *12/streams*, funcţia:
  - *standardFileIO();*

Biblioteca standard pentru fluxuri de I/O (Standard IO stream library)

- O bibliotecă orietată obiect care oferă functionaţităţi pentru *stream*-urile de I/O

- *Stream*
  - O abstractizare care reprezintă un dispozitiv cu ajutorul căruia se efectuează operaţii de I/O
  - Poate fi repzentat ca o sursă sau destinaţie formată din-un număr neprecizat de caractere

- Fişiere componente ale bibliotecii:
  - <ios>, <istream>, <ostream>, <streambuf> şi <iosfwd>
    - clasele de bază
  - <iostream>
    - Declară obiecte folosite în comunicarea dintre intrarea şi ieşirea standard
    - include obiecte precum *cin* şi *cout*
  - <fstream>
    - Defineşte clasele pentru lucrul cu fişiere
  - <sstream>
    - Pentru manipularea stringurilor din *stream*-uri
  - <iomanip>
    - Modifică *flag*-urile interne şi optţiunile de formatare

Elementele bibliotecii IOS

- Clase
  - Clase template de bază
  - Instanţieri ale claselor template
    - *Narrow-oriented*
    - *Wide-oriented*
- Obiecte standard
- Tipuri
- Manipulatori

De ce *streams*?

- Citirea şi afişarea în C
  - *scanf*
  - *printf*

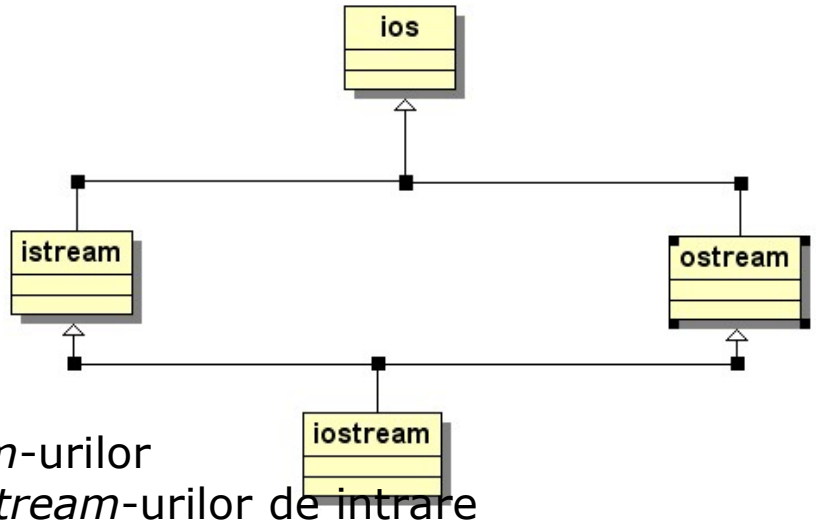
  - Nu pot fi extinse
  - Tratează doar tipurile de date de bază (int, char, float, char*, etc)

- ➔ funcţii pentru *stream*-uri

Stream-uri

- *Stream* = un obiect care transportă şi formatează caractere:
  - de la o sursă:
    - tastatură(stdin),
    - fişier sau
    - zonă de memorie
  - la o destinaţie:
    - ecran (stdout),
    - fişier sau
    - zonă de memorie.
  - Caracterele sunt procesate în mod serial (unele dupa altele)

Ierarhia de clase



- *ios* → comportamentul general al *stream*-urilor
- *istream* → comportamentul general al *stream*-urilor de intrare
- *ostream* → comportamentul general al *stream*-urilor de ieşire
- *iostream = istream + ostream*

- Obiecte:
  - *cin* – un obiect de tip *istream* (*stream*-ul standard de intrare)
  - *cout* – un obiect de tip o*stream* (*stream*-ul standard de ieşire)
  - cerr – un obiect de tip o*stream* (*stream*-ul standard de eroare)
  - clog – un obiect de tip o*stream* (*stream*-ul standard de *logging*)

- Operatori
  - << (inserter)
  - >> (extracter)
    - Exclude spaţiile atunci când *flag*-ul *ios::skipws* este setat (implicit)

Starea unui stream

Setarea *flag*-ului unui *stream*: *streamObj.setstate(flag)*

```
void setFlag(){
        ofstream os;
        os.setstate(ios::failbit);
}
```

Flag	**Explicaţie**
fail	Date invalide
badbit	Eroare fizică
goodbit	OK
eofbit	A fost detectat caracterul sfârşit de fişier (EOF)

Starea unui *stream*

- Verificarea stării unui *stream*

bool fail()
bool good()
bool eof()
bool bad()

```
void checkTheFlag(){
        ofstream os;
        if (os.good()){
                //use os
        }
}
```

Starea unui *stream*

- Resetarea tuturor *flag*-urlor: *streamObj.clear()*
  - Toate *flag*-urile sunt resetate
  - Funcţia *good()* va returna *true*

```
void resetFlags(){
        ofstream os;
        os.setstate(ios::failbit);
        os.clear();
        bool b = os.good();              //b = true
}
```

Stream-uri - exemple

- A se consulta exemplul din directorul *12/streams*, funcţiile:
  - *setFlag();*
  - *checkTheFlag();*
  - *resetFlags();*
  - *dataProcessing();*

Intrări orientate pe linii

- Problemă
  - *cin*>> - citirea se opreşte la primul spaţiu
- Soluţia
  - *istreamObj.get()*
  - *istreamObj.getline()*
  - *istreamObj.ignore()*

istreamObj.get()

- Similar lui >>, dar cu 2 excepţii:
  - Include şi spaţiile
  - e mai puţin probabil ca *stream*-ul de ieşire să fie curăţat

- 3 versiuni:
  - ***istreamObj.get()***
    - citeşte primul caracter citit din *stream* şi în returnează
    - celelalte caractere rămân în *stream*

  - ***istreamObj.get****(bufferAdress, maxNoOfElem)*
    - citeşte din *stream* în *buffer* maximum *maxNoOfElem* caractere
    - celelalte caractere rămân în *stream*

  - ***istreamObj.get****(bufferAdress, maxNoOfElem, delim)*
    - citeşte din stream în *buffer* maximum *maxNoOfElem* caractere până la întâlnirea primei apariţii a delimitatorului *delim*
    - celelalte caractere rămân în *stream*
    - caracterul de terminare rămâne în *stream*

istreamObj.getline(…)

- Similar cu *get*, dar cu 1 excepţie:
  - Elimină caracterul de terminare din stream (pentru ultima versiune)

- 2 versiuni:
  - ***istreamObj.get****(bufferAdress, maxNoOfElem)*
    - citeşte din *stream* în *buffer* maximum *maxNoOfElem* caractere
    - celelalte caractere rămân în *stream*
  - ***istreamObj.get****(bufferAdress, maxNoOfElem, delim)*
    - citeşte din *stream* în *buffer* maximum *maxNoOfElem* caractere până la întâlnirea primei apariţii a delimitatorului *delim*
    - celelalte caractere rămân în *stream*
    - elimină din *stream* caracterul de terminare

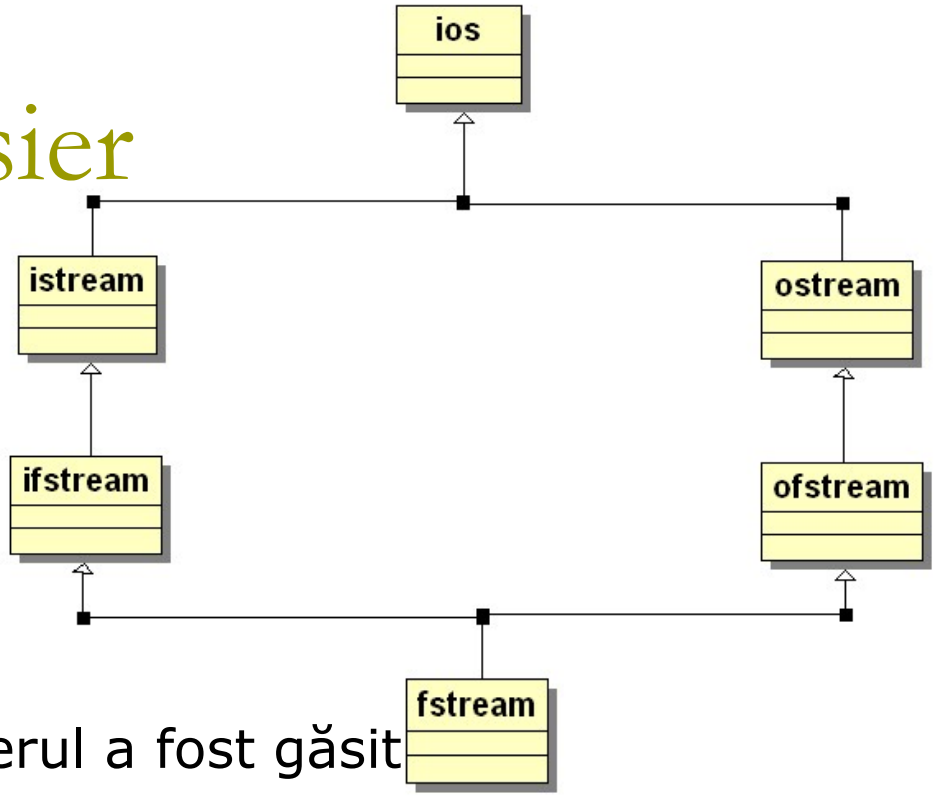
istreamObj.ignore(…)

- ***istreamObj.ignore****(count, delim)*
  - extrage şi descarcă din *stream* până la *count* caractere
  - extragerea se oprezte dacă
    - delimitatorul *delim* este extras
    - sfărşitul de *stream* este întâlnit
  - caracterul de delimitare este extras din *stream*

Exemplu

- A se consulta exemplul din directorul *12/streams*, funcţiile:
  - *getExample();*
  - *getlineExample();*

Stream-uri de tip fişier



- *ifstream*
  - Trebuie verificat dacă fişierul a fost găsit
- *ofstream*
  - Fişierul trebuie închis

- A se consulta exemplul din directorul *11/streams*, funcţia
  - *iofstreamExample();*

Deschiderea unui *stream* de tip fişier

- Cum?
  - Utilizarea constructorului
  - Utilizarea funcţiei *open(...)*
  - În ambele cazuri, există 2 posibilităţi:
    - Folosirea numelui fişierului
    - Folosirea numelui fişierului şi a modului de deschidere (*flag*)

Flag	**Explicaţie**
ios::in	Deschidere pentru citire (valoare implicită pentru *ifstream*)
ios::out	Deschidere pentru scriere (valoare implicită pentru *ofstream*)
ios::ate	Setarea indicatorului de poziţie la sfârşitul *stream*-ului în momentul deschiderii (*at end*)
ios::app	Setarea indicatorului de poziţie la sfârşitul *stream*-ului înainte de orice operaţie de scriere (*append*)
ios::binary	Considerarea binară a *stream*-ului (în loc de text)
ios::trunc	Curăţarea *stream*-ului ➔ la deschidere, *stream*-ul va avea lungimea 0

Închiderea unui *stream* de tip fişier

- Cum?
  - Folosirea (automată) a destructorului
  - Folosirea funcţiei *close()*

Stream-uri tampon (*stream buffering*)

- rdfbuf(…)
  - Get/set stream-ul tampon (*stream buffer*) asociat
  - *streambuf** ***rdbuf*** *( )*
    - Returnează obiectul de tip *stream buffer* asociat *stream-ului*
  - *streambuf** ***rdbuf*** *(streambuf* sb)*
    - Asociază stream-ul cu *sb* şi returnează obiectul de tip *stream buffer* asociat anterior *stream*-ului

- A se consulta exemplul din directorul *12/streams*, funcţia
  - *streamBuffering()*

Poziționarea într-un fișier

- Poziţia absolută
  - *ostream*:
    - ***tellp()*** → returnează poziţia absolută
    - ***seekp()*** → setează poziţia absolută
  - *istream*:
    - ***tellg()*** → returnează poziţia absolută
    - ***seekg()*** → setează poziţia absolută
- Poziţia relativă

Flag	**Explicaţie**
ios::bag	Poziţionare la început de fişier
ios::cur	Poziţionare la poziţia curentă din fişier
ios::end	Poziţionare la sfârşit de fişier

Poziționarea într-un fişier

- ***seekp****(0, ios::beg)*
  - Deplasare cu 0 caractere relativ faţă de începutul fişierului
- ***seekp****(-2, ios::end)*
  - Deplasare cu 2 caractere înainte de sfârşitul fişierului
- ***seekp****(10, ios::cur)*
  - Deplasare cu 10 caractere faţă de poziţia curentă din fişier

Formatarea intrărilor

- 2 metode
  - Cu ajutorul funcţiilor membre
  - Cu ajutorul manipulatorilor

- Bazată pe câteva *flag*-uri:
  - *Flag*-uri on/off – prin folosirea metodelor *setf/unsetf*
  - *Flag*-uri care lucrează în grup

Formatarea intrărilor

- *Flag*-uri on/off

Flag	**Explicaţie**
ios::skipws	Ignorarea spaţiilor
ios::showbase	Afişarea bazei de numerţie
ios::showpoint	Afişarea punctului zecimal
ios::showpos	Afişarea semnului
ios::uppercase	Generarea de litere mari
ios::unitbuf	Curăţarea *buffer*-ului după încărcare

Formatarea intrărilor

- *Flag*-uri grupate
  - Recomandare – folosirea doar a unui flag dintr-un grup

Flag		**Explicaţie**
ios::basefield	*ios::dec*	Folosirea bazei 10
	ios::hex	Folosirea bazei 16
	ios::oct	Folosirea bazei 8
ios::floatfield	*ios::fixed*	Folosirea notaţiei zecimale
	ios::scientific	Folosirea notaţiei ştiinţifice
ios::adjustfield	*ios::right*	Ajustarea ieşirii la dreapta
	ios::left	Ajusatrea ieşirii la stânga
	ios::interval	Ajustarea unui câmp prin inserarea de caractere la o anumită poziţie internă

Formatarea intrărilor

- Get/set pentru dimensiune
  - *streamsize* ***widht****()*
    - Returnează numărul minim de caractere care trebuie afişate pentru o anumită reprezentare
  - *streamsize* ***widht****(streamsize wide)*
    - Setează o nouă dimensiune

- Get/set pentru caracterul de umplere
  - *char* ***fill****()*
    - Returnează caracterul de umplere
  - *char* ***fill****(char filch)*
    - Setează caracterul de umplere la noua valoare (*fillch*) şi îl retunrează

- Get/Set pentru precizia punctului zecimal
  - *streamsize* ***precision****( )*
    - Returnează precizia (nr de zecimale)
  - *streamsize* ***precision****(streamsize prec)*
    - Setează precizia la noua valoare *prec*

Exemplu

- A se consulta exemplul din directorul *12/streams*, funcţia:
  - *formatFlags();*

Manipulatori

- Funcţii speciale pentru a fi folosite în conjuncţie cu operatorii de inserare (<<) şi extragere (>>) ai obiectelor de tip *stream*
- Echivalenţi cu flag-urile şi funcţiile anterioare

Funcţie	**Explicaţie**
ios::skipws() *Ios::noskipws()*	Ignoră spaţiile
ios::showbase() *ios::noshowbase()*	Afişează baza numerică
ios::showpoint() *ios::noshowpoint()*	Afşează punctul zecimal
ios::showpos() *ios::noshowpos()*	Afişează semnul
ios::uppercase() *ios::nouppercase()*	Generează litere mari
ios::unitbuf() *ios::nounitbuf()*	Curăţă *buffer*-ul după inserări

Manipulatori

Funcţie		**Explanation**
ios::basefield	*ios::dec()*	Utilizarea bazei 10
	ios::hex()	Utilizarea bazei 16
	ios::oct()	Utilizarea bazei 8
ios::floatfield	*ios::fixed()*	Utilizarea notaţiei zecimale
	ios::scientific()	Utilizarea notaţiei ştiinţifice
ios::adjustfield	*ios::right()*	Ajustarea ieşirii la dreapta
	ios::left()	Ajustarea ieşirii la stânga
	ios::interval()	Ajustarea câmpului prin inserarea unor caractere a o anumită poziţie internă

Manipulatori de I/O

- *ostream& function(istream &is)*
- *ostream& function(ostream &os)*

Tip	**Funcţie**	**Explicaţie**
Input	*ws*	Extrage spaţiile
Output	*endl*	Inserează linie nouă şi curăţă *buffer*-ul
	ends	Inserează caracterul nul
	flush	Curăţă *stream buffer*-ul

Manipulatori cu argumente (parametrizați)

- Necesită includerea fişierului *header* <iomanip>

Funcţie	**Explicaţie**
setiosflags(mask)	Setează *flag*-urile de formatare specificate de parametrul*mask*
resetiosflags(mask)	De-setează *flag*-urile de formatare specificate de parametrul*mask*
setbase(base)	Setează *flag*-ul de formatare pentru bază la valoarea *base*
setfill(c)	Setează caracterul de umplere la valoarea *c*
Setprecision(p)	Seteză precizia operaţiilor de ieşire la valoarea *p*
setw (n)	Setează dimensiunea pentru următoarea operaţie de inserare la valoarea lui *n*

Exemplu

- A se consulta exemplul din directorul *12/streams*, funcţia:
  - *manipulators();*