

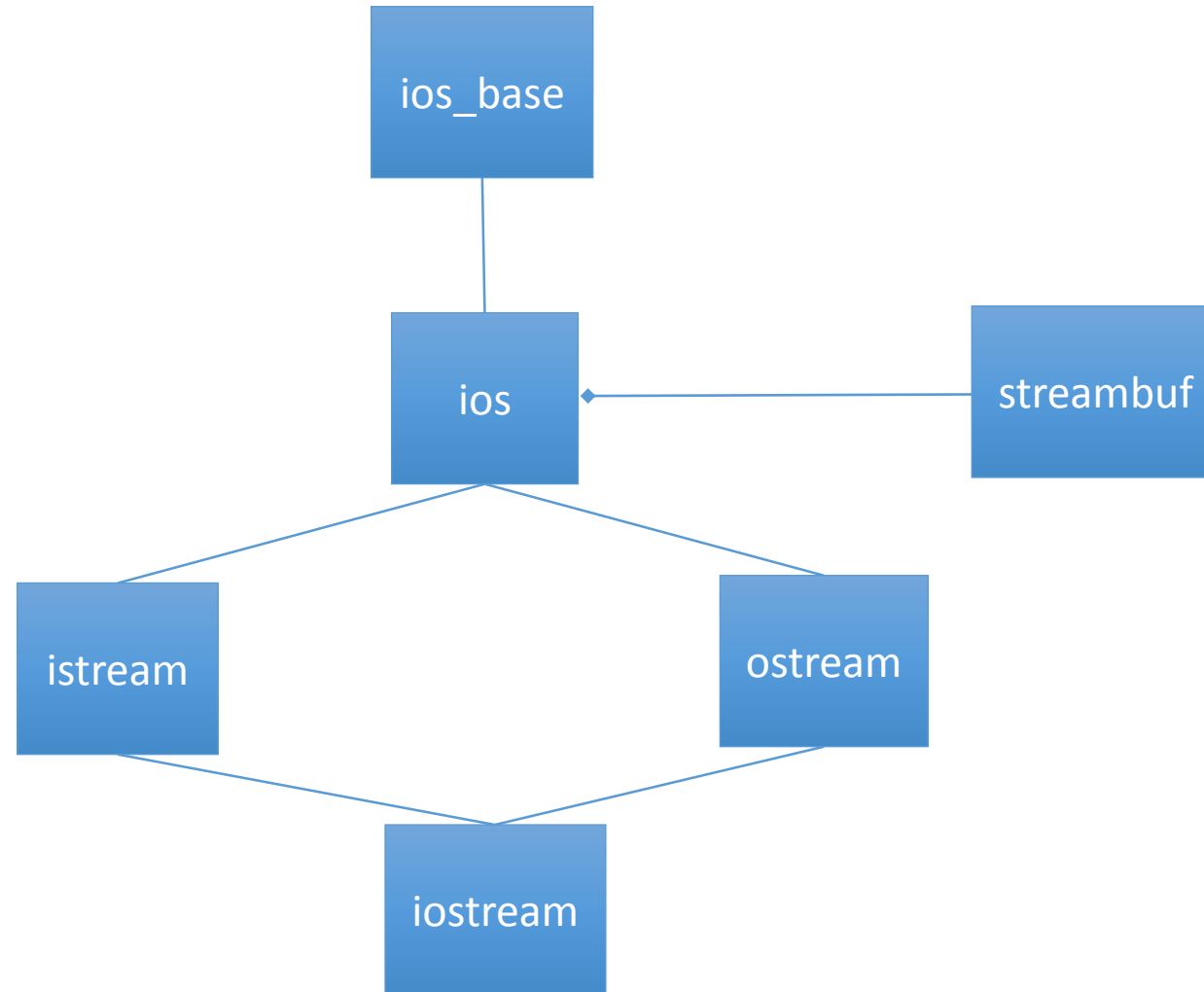
Programare orientată obiect

Cursul 9

Sumar

- Intrări/ieșiri în C++ (3)
 - Intrări/ieșiri standard
 - Controlul operațiilor de intrare/ieșire
 - Formatarea conținutului
 - Manipulatori de format
 - Metode
 - Instrări/ieșiri nestandard
 - Fișiere
 - Memorie

I/O C++ standard



I/O C++ standard

- Operații de nivel ridicat
 - Formate
 - Tipuri de date fundamentale (implicit)
- Nivel scăzut
 - Neformate
 - La nivel de octet/caracter

istream – alte metode

- Citirea unui caracter
 - **get()**
 - **get(char &)**
- Citirea unui șir de caractere, inclusiv spații
 - **get(char *, int)** sau **get(char *, int, char)**
 - **getline(char *, int)**
- Citirea unui șir de octeți (neformatată)
 - **read(char *, int)**

istream – alte metode

- Returnarea caracterului următor din flux (fără extragere)
 - int **peek()**
- Inserarea unui caracter în flux
 - **putback(char)**
- Obținerea numărului de caractere citite
 - int **gcount()**
- Extragerea și eliminarea caracterelor din flux
 - **ignore(int=1, char=EOF)**

ostream – alte metode

- Afișarea unui caracter
 - **put(char)**
 - Returnează o referință de tipul apelatorului
- Afișarea neformatată a unui șir de octeți
 - **write(char *, int)**
- Golire buffer
 - **flush()**

Controlul operațiilor de intrare

- **Indicatori de stare:**
 - `ios::failbit` – tipuri incompatibile
 - `ios::badbit` – operație eșuată
 - `ios::goodbit` – operație reușită
 - `ios::eofbit` - sfârșit de fișier

Controlul operațiilor de intrare

- **fail()** – returnează rezultatul ultimei operații de intrare (succes/eșec)
 - *failbit* sau *badbit* setați
- **bad()** – returnează rezultatul ultimei operații de intrare (succes/eșec)
 - *badbit* setat
- **eof()** – returnează true (1) dacă este sfârșit de fișier
- **clear()** – resetează indicatorii de eroare (toți sau cei specificați)

Controlul operațiilor de intrare

```
int nr;
```

```
cin>>nr;
```

```
if (cin.fail())
```

```
{  
    //prelucrare eroare  
}
```

```
int nr;
```

```
if (!(cin>>nr))
```

```
{  
    //prelucrare eroare  
}
```

Controlul formatului

- Afișarea valorilor numerice și logice
- Alinierea conținutului
 1. Metode specializate ale claselor de I/O
 2. Manipulatori (funcții) specializați
 3. Metode și manipulatori cu indicatori de format

1. Metode (format)

- Setare/obținere precizie de afișare a numerelor reale
 - int **precision**(int)
 - int **precision**()
- Setare/obținere dimensiune minimă a unui șir (prin completare)
 - int **width**(int)
 - int **width**()
- Setare/obținere caracter de umplere
 - char **fill**(char)
 - char **fill**()
- Exemple:
 - **cout.precision(2);**
 - **cout.width(20);**
 - **cout.fill('*');**

2. Manipulatori

- Funcții care modifică parametrii de format ai fluxurilor de intrare/ieșire
- Se utilizează cu obiectele de intrare/ieșire și operatorii << și >>:
 - obFlux<<...<<manipulator<< ...
 - obFlux>>...>>manipulator>> ...
- Tipuri
 - Fără parametri, de tip setat/resetat
 - Cu parametri
 - Specifici
 - Indicatori
- Formatul ramîne activ pînă la următoarea schimbare
 - Excepție la width
- #include <iomanip>
 - Pentru manipulatorii cu parametri

Manipulatori

- **hex, dec, oct** – afișare baza 16, 10 și 8
- **showbase/noshowbase** – afișarea bazei numerice
- **uppercase/nouppercase** – afișare litere mari
- **fixed** – numărul de zecimale este fix (se afișează 0)
- **scientific** – se utilizează forma științifică
- **showpos/noshowpos** – afișarea semnului
- **showpoint/noshowpoint** – afișarea punctul zecimal
- **boolalpha / noboolalpha** – afișarea valorilor logice

Manipulatori

- Precizia la afișarea unui număr real
 - **setprecision(int)**
- Stabilirea bazei utilizate (8, 10 sau 16)
 - **setbase(int)**
- Exemple:
 - `cout<<setprecision(2)<<x<<endl;`
 - `cout<<setprecision(3); cout<<x;`
 - `cin>>setbase(16);`

Example

- `bool b;`
- `cin>>boolalpha>>b;`
- `cout<<boolalpha<<b;`
- `cout<<noboolalpha<<b;`

Manipulatori (intrare)

- Extragere spații albe
 - **ws**
- Ignorare spații albe
 - **skipws** (implicit)/**noskipws**
- Exemple:
 - `char c;`
 - `cin>>noskipws>>c;`

Manipulatori (ieşire)

- Golire buffer
 - **flush**
- Linie nouă și golire buffer
 - **endl**
- Inserarea caracterului nul (terminator de șir)
 - **ends**

Manipulatori

- Stabilirea dimensiunii minime a unui șir (prin completare); la intrare, limitează numărul de caractere citite:
 - **setw**(int)
- Stabilirea caracterului utilizat pentru completare
 - **setfill**(char)
 - implicit este spațiu
- Alinierea conținutului/completare
 - **right**
 - dreapta (implicit)
 - **left**
 - stînga
 - **internal**
 - dreapta (text),
 - stînga (semn, identificator bază) și dreapta (valorile numerice)

Manipulatori

- Exempolu
 - `int x = -200;`
 - `cout<<setfill('*');`
 - `cout<<setw(20)<<left<<x<<endl;`
 - `cout<<setw(20)<<right<<x<<endl;`
 - `cout<<setw(20)<<internal<<x<<endl;`