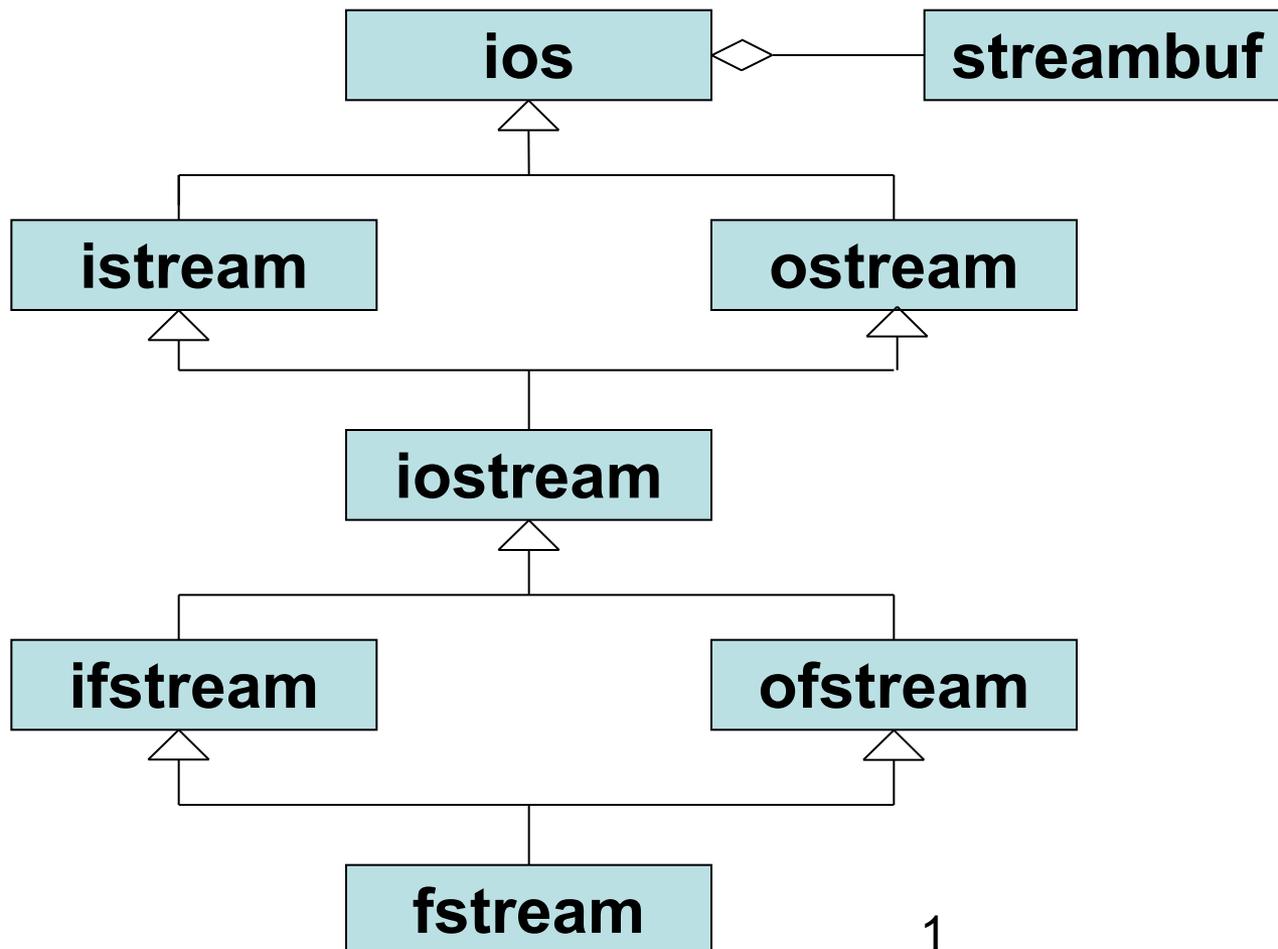


12 Библиотека ввода/вывода C++

12.1 Стандартные потоки

Стандартные потоки:

`cin`, `cout`, `clog`, `cerr`



Библиотека ввода/вывода C++ (2)

Операции ввода/вывода делятся на две группы:

- а) операции с преобразованием из внутреннего формата в символьный – осуществляется переопределенными “<<” и “>>”;
- б) операции без преобразования – осуществляется специальными функциями.

12.2 Операции «извлечение» и «вставка»

Операции переопределены для:

**char, signed и unsigned short, int, long, float, double,
long double, char *(строка), void *(адрес)**

Примеры:

а) **cout<<“Input integer:”;**

б) **cin>>a;**

в) **cout<<2+a+8;**

г) **cout<<“String:”<<str<<endl;**

д) **cout<<(a<<2);**

е) **cout<<(a=2);**

12.3 Управление потоками ввода/вывода

Форматирование ввода/вывода. Флаги

enum

```
{ skipws = 0x0001, // пропустить пробелы при вводе
  left   = 0x0002, // выполнять по левой гр. при выводе
  right  = 0x0004, // выполнять по правой гр. при выводе
  interval=0x0008, // дополнить пробелами при выводе
  dec    = 0x0010, // преобразовать в десятичную с/с
  oct    = 0x0020, // преобразовать в восьмиричную с/с
  hex    = 0x0040, // преобразовать в шестнадцатир. с/с
  showbase=0x0080, // показывать основание с/с при выводе
  showpoint=0x0100, // показывать дес. точку при выводе
  uppercase=0x0200, // вывод шестн. цифр в верхнем р-ре
  showpos = 0x0400, // выводить + перед полож. числами
  scientific=0x0800, // вывод в формате с плав. точкой
  fixed    = 0x1000, // вывод в формате с фикс. точкой
  unitbuf  = 0x2000, // стереть все потоки после вставки
  stdio    = 0x4000}; //стереть после вставки stdin, stdout
```

Флаги (2)

```
class ios
{private:
    long x_flags;      // флаги
    int x_width;      // ширина поля вывода
    int x_precision; // число цифр др. части
    int x_fill; ...} // символ-заполнитель
```

Для работы с этими полями используются специальные методы:

```
flags(), precision(), setf(), width()
```

Пример:

```
cout.setf(ios::uppercase);
cout.setf(ios::hex, ios::dec&oct&hex);
```

6.4 Манипуляторы

1) Манипуляторы без параметров (**iosream.h**):

ostream & <Имя> (ostream & Os)

dec – десятичная система счисления;

hex – шестнадцатеричная система счисления;

oct – восьмеричная система счисления;

ws – удаление из входного потока пробелов и знаков табуляции;

endl – добавление маркера «конец строки» + вывод из буфера;

flush – вывод из буфера.

2) Манипуляторы с параметрами (**iosmanip.h**):

setbase(int n) – установка системы счисления (0, 8, 10, 16);

setprecision(int n) – определяет количество дробных цифр;

setw(int n) – определяет минимальную ширину поля вывода.

Пример:

```
cout<<setw(6)<<setprecision(2)<<b;
```

12.5 Переопределение извлечения и вставки для типов пользователя

Переопределяются как функции -операции вне класса :

```
ostream & operator<<(ostream &out,<Тип> <Имя>)
{ ...
  out<<...
  return out;
}
```

```
istream & operator>>(istream &in,<Тип> &<Имя>)
{ ...
  in>>...
  return in;
}
```

Пример переопределения извлечения и вставки (Ex5_15)

```
#include "stdafx.h"
#include <iostream.h>
#include <iomanip.h>
class TVector
{ private: int x,y,z;
  public:  TVector() {}
         friend ostream& operator<<(ostream &stream,
                                     TVector obj);
         friend istream& operator>>(istream &stream,
                                     TVector& obj);
};
```

Пример переопределения извлечения и вставки

```
ostream& operator<<(ostream &stream, TVector
    obj)
{ stream << "Value :";
  stream << setw(5)<< obj.x<<" , "<< obj.y;
  stream << " , " << obj.z << "\n";
  return stream;
}

istream& operator>>(istream &stream,
    TVector& obj)
{ cout<<"Input value:";
  return stream >> obj.x >> obj.y >> obj.z;
}
```

Тестирующая программа

```
int main(int argc, char* argv[])
{   TVector A,B;
    cin>>A>>B;
    cout<<A<<B;
    return 0;
}
```