

гл.ас. д-р.Нора Ангелова

# Потоци

# Входно-изходни операции

Вече познаваме такива 😊

- В C++ входно-изходните операции се реализират чрез потоци.

```
cout <<  
cin >>
```

# Поток

- Редица от байтове

Реализация:

- Обект на клас, който реализира обмен на данни между източника и приемника



# Поток

---

Входни операции

- От устройството към ОП

Изходни операции

- От ОП към устройството

Устройство може да бъде - клавиатура, диск и др.

# Входен/Изходен Поток

Входен поток

- Поток, който е свързан с определен източник на данни.

Пример:  
`cin`

Изходен поток

- Поток, който е свързан с определен приемник на данни.

Пример:  
`cout`

# Поток

---

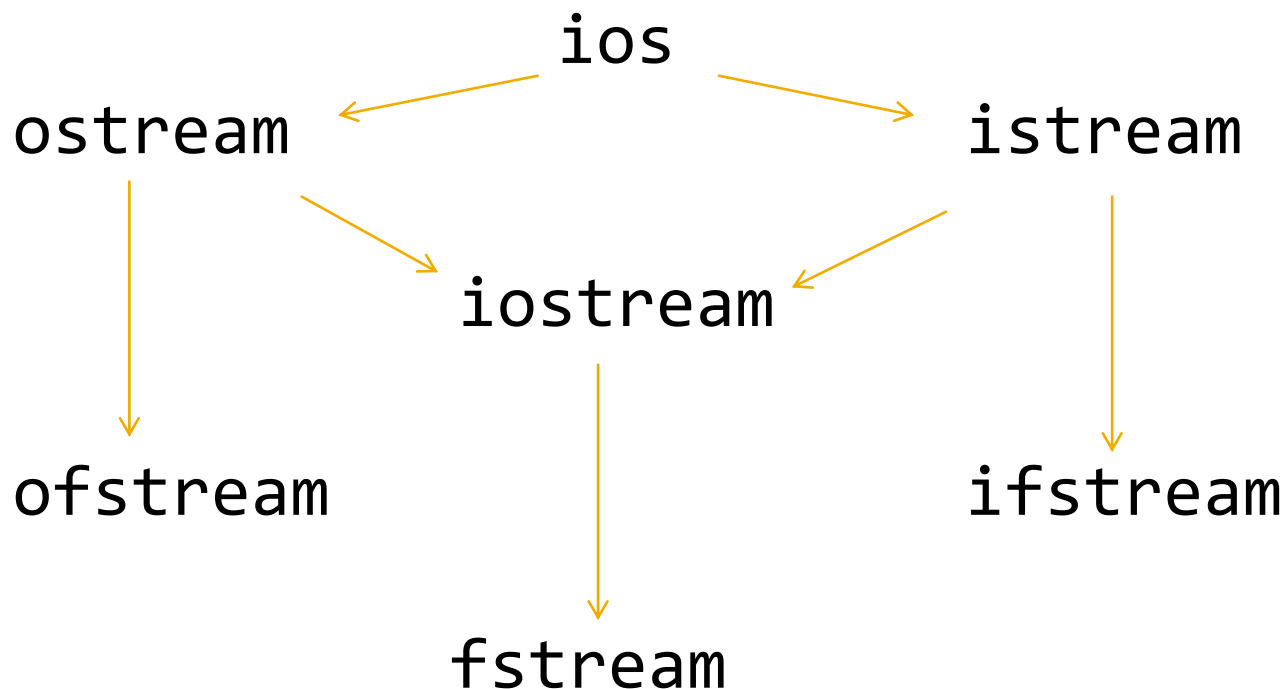
Библиотека – `iostream`

- Съдържа множество от класове

# Поток

Библиотека – `iostream`

- Съдържа множество от класове



# Поток

- `cout` – обект от клас `ostream`;
- `cin` – обект от клас `istream`;
- `iostream` – съдържа дефиниции на стандартните потоци (`cin`, `cout`, `cerr` и `clog`);

`cerr` – свързан е към стандартното устройство за грешки. Осигурява небуфериран изход на съобщенията за грешки (съобщенията се извеждат веднага).

`clog` – свързан е към стандартното устройство за грешки. Осигурява буфериран изход на съобщенията за грешки (съобщенията са натрупват в буфер).



# Входно/изходни оператори

```
istream& operator>>(istream& is, <Type>& t) {
```

- is – обект на класа istream;
- t – обект на клас Type, за който се дефинира оператора;

```
ostream& operator<<(ostream& os, <Type>& t)
```

- os – обект на класа ostream;
- t – обект на клас Type, за който се дефинира оператора;

# Входно/изходни оператори

```
ostream& operator<<(ostream& os, Paper& p) {  
    os << p.id << ... ;  
    return os;  
}
```

```
istream& operator>>(istream& is, Paper& p) {  
    is >> p.id >> ... ;  
    return is;  
}
```

*\*operator >> и << са предефинирани за id*

*\*имаме ли достъп до член-данните на Paper*

# Входно/изходни оператори

```
class Paper {  
  
    friend ostream& operator<<(ostream&, Paper&);  
    friend istream& operator>>(istream&, Paper&);  
    ...  
  
    int id;  
};
```

# Член-функции за вход/изход

## ostream

- `ostream& put(char)` – записва символа в изходния поток.

Пример:

```
cout.put( 'A' ).put( 'B' ).put( 'C' ).put( 'D' );
```

Резултат: ABCD

# Член-функции за вход/изход

## ostream

- ostream& write(const char\* str, streamsize size)

str – низ;

size – брой на символите, които ще бъдат записани;

Пример:

```
cout.write("ABCD", 3).write("ABCD", 2);
```

Резултат: ABCAB

# Член-функции за вход/изход

## istream

- `istream& get(char& ch)` – извлича един символ и го свързва с променливата `ch`.

Пример:

```
char c1, c2, c3, c4;
```

```
cin.get(c1).get(c2).get(c3).get(c4);
```

# Член-функции за вход/изход

## istream

- `istream& get(char* str, streamsize size);`
  - `istream& get(char* str, streamsize size, char delim);`
- `str` – низ;  
`size` – максимален брой на символите, които ще бъдат извлечени;  
`delim` – символ, който се нарича разделител.

Пример:

```
char str[10];  
cin.get(str, 10, '.');  
// abcd.defg.1234
```

Резултат:

`str: abcd`

# Член-функции за вход/изход

## istream

- `istream& get(char* str, streamsize size);`
  - `istream& get(char* str, streamsize size, char delim);`
- `str` – **НИЗ**;  
`size` – максимален брой на символите, които ще бъдат извлечени;  
`delim` – символ, който се нарича разделител.

Пример:

```
char str[10], str2[10];  
cin.get(str, 10, '.').get(str2, 10, '?');  
// abcd.defg.1234
```

Резултат:

`str: abcd`      `str2: .defg.123`



# Член-функции за вход/изход

## istream

- `istream& getline(char* str, streamsize size);`
- `istream& getline(char* str, streamsize size, char delim);`

`str` – низ;

`size` – максимален брой на символите, които ще бъдат извлечени;

`delim` – символ, който се нарича разделител.

Пример:

```
char str[10], str2[10];  
cin.getline(str, 10, '.').getline(str, 10, '.');  
// abcd.defg.1234
```

Резултат:

str: abcd      str2: defg

# Член-функции за вход/изход

## istream

- `istream& read(char* str, streamsize size);`

`str` – **массив**;

`size` – максимален брой на символите, които ще бъдат извлечени;

Пример:

```
char str[10];  
cin.read(str, 10);  
// 0123456789abc
```

Резултат:

str: 0123456789

# Член-функции за вход/изход

## istream

- `int peek();`

Връща ASCII кода на поредния символ, но не го извлича от входния поток.

- `istream& putback(char ch);`

Връща символ в потока.

Пример:

```
char c1, c2;  
cin.get(c1).get(c2);  
cin.putback('9').putback('5');  
cin.get(c1).get(c2);  
cout << c1 << c2;
```

Резултат:

59

# Член-функции за вход/изход

Пример:

```
ifstream fin("file1", ios::in);  
ofstream fout("file2", ios::out);
```

```
while (fin.get(ch)) fout << ch;  
while (fin.get(ch)) fout.put(ch);
```

Копира дословно файла.

```
while (fin >> ch) fout << ch;
```

Не копира интервали, нов ред, табулация.

# Състояние на поток

- Задава се чрез множество от битове.
- Битовете изпълняват ролята на флагове.
- Флаговете са константи от изброим тип – `iostate`.

`ios::goodbit` – операциите са изпълнени успешно.

`ios::eofbit` – достигнат е край на файла (**активира и `failbit`**).

`ios::failbit` – последната входно/изходна операция е неуспешна.

`ios::badbit` – изпълнена е невалидна операция и има загубена информация.

# Състояние на поток

- `iostate rdstate() const` – връща текущото състояние на потока
- `void clear(iostate fl = goodbit)` // параметър по подразбиране  
Модифицира състоянието на потока.
- `bool good() const` – връща истина, ако съответния флаг е активиран;
- `bool eof() const` – връща истина, ако съответния флаг е активиран;
- `bool fail() const` – връща истина, ако съответния е възникнала грешка. Потокът може да се използва, ако флагът се нулира.
- `bool bad() const` – връща истина, ако съответния флаг е активиран;

# Състояние на поток

```
#include <iostream>
using namespace std;
int main() {
    double x;
    cout << "x: "; // 3.5 || 'a'
    cin >> x; cout << x << endl;
    cout << "good(): " << cin.good() << endl
         << "bad(): " << cin.bad() << endl
         << "fail(): " << cin.fail() << endl
         << "eof(): " << cin.eof() << endl;
    return 0;
}
```

# Състояние на поток

```
char ch1, ch2, ch3;
cin.get(ch1).get(ch2).get(ch3);
cout << ch1 << ch2 << ch3 << endl;
cin.putback('1').putback('2').putback('3').putback('k');

cout << cin.rdstate() << endl; // 4 - badbit
cin.clear(); // възстановява cin, но има загуба на

// входни данни
cin.get(ch1).get(ch2).get(ch3);
cout << ch1 << ch2 << ch3 << endl;
```



Край