

ПОТОК

доц. д-р Нора Ангелова

ВХОДНО-ИЗХОДНИ ОПЕРАЦИИ

Вече познаваме такива 😊

Примери:

```
cout << ...;
```

```
cin >> ...;
```

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

ПОТОК

- **Дефиниция** - Редица от байтове, които се движат в една посока.

Представяват абстрактни канали за данни, до които достъпът се осъществява последователно.

Предоставят механизъм за четене и писане на поредица байтове от и към устройства за съхранение или пренос на данни.

- **Реализация** - обект на клас, който реализира обмен на данни между източника и приемника.



ВХОДНО-ИЗХОДНИ ОПЕРАЦИИ

- ◉ **Входни операции** - потокът предава данни от устройството към оперативната памет.



- ◉ **Изходни операции** - потокът предава данни от оперативната памет към устройството.



* *Устройство може да бъде клавиатура, диск и др.*

ВХОДЕН ПОТОК

- ◉ **Входен поток** - поток, който е свързан с определен източник на данни.

Пример:

`cin`

ИЗХОДЕН ПОТОК

- **Изходен поток** - поток, който е свързан с определен приемник на данни.

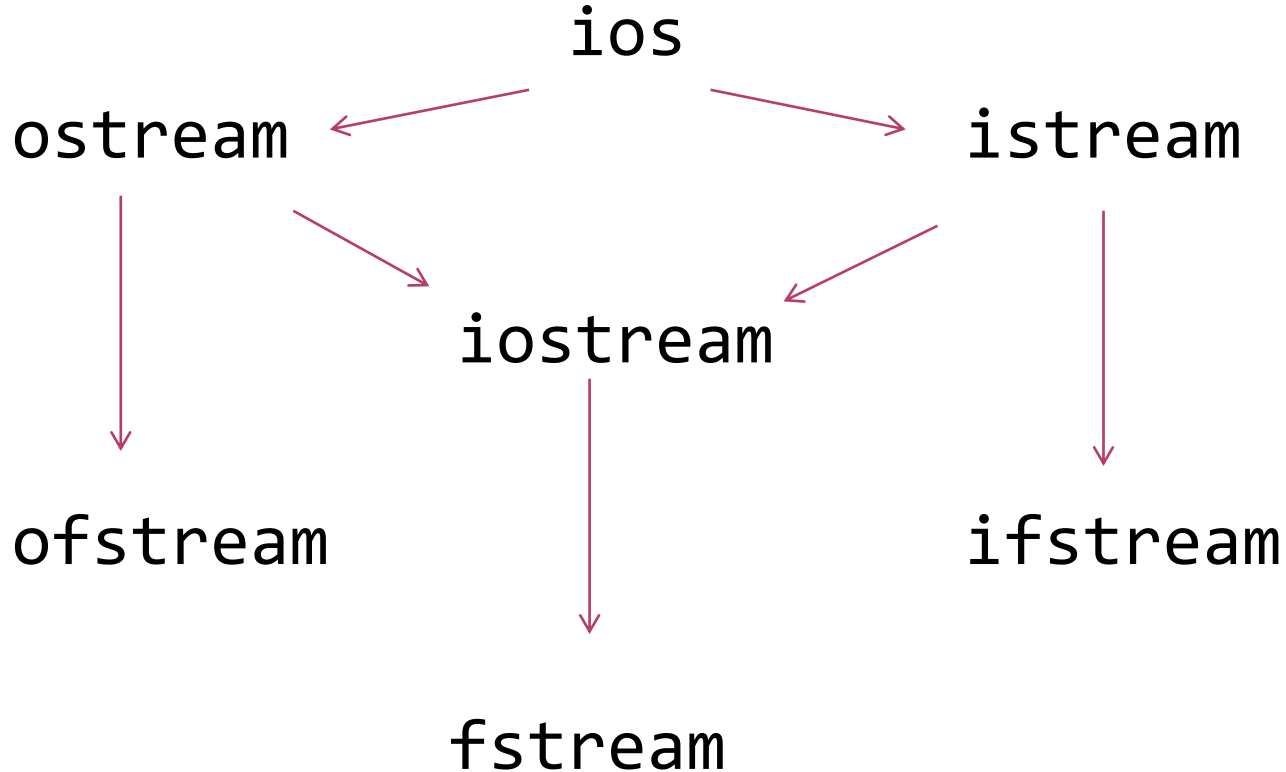
Пример:

cout

ПОТОК

Библиотека - `iostream`

- Съдържа множество от класове.



ПОТОК

`iostream` - съдържа дефиниции на стандартните потоци (`cin`, `cout`, `cerr` и `clog`);

- `cout` - обект от клас `ostream`;
- `cin` - обект от клас `istream`;
- `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).

ПОТОК

- **Указатели**

`ifstream` или `istream` - **get** указател, който реферира елемента, който ще се прочете при следващата входна операция.

`ofstream` или `ostream` - **put** указател, който реферира мястото, където ще се запише следващият елемент.

ЧЛЕН-ФУНКЦИИ ЗА ВХОД/ИЗХОД

◉ ostream

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

Пример:

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

Резултат:

ПОТОК

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:

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

Резултат:

ПОТОК
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;  
std::cin.get(c1).get(c2).get(c3).get(c4);
```

Резултат:

ПОТОК

ABCAB

BCAB

CAB

AB

B

`c1 == 'A'` `c2 == 'B'` `c3 == 'C'` `c4 == 'A'`

ЧЛЕН-ФУНКЦИИ ЗА ВХОД/ИЗХОД

◎ istream

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

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

str - [НИЗ](#);

[size - 1](#) - максимален брой на символите, които ще бъдат извлечени;

delim - символ, който се нарича разделител.

Пример:

```
char str[10];
```

```
std::cin.get(str, 10, '.');
```

Резултат:

ПОТОК

abcd.defg.1234

str - abcd

ЧЛЕН-ФУНКЦИИ ЗА ВХОД/ИЗХОД

◉ istream

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

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

str - [НУЗ](#);

[size - 1](#) - максимален брой на символите, които ще бъдат извлечени;

delim - символ, който се нарича разделител.

Пример:

```
char str[10], str2[10];
```

```
std::cin.get(str, 10, '.').get(str2, 10, '.');
```

Резултат:

ПОТОК

abcd.defg.1234

str - abcd;

str2 - “”;

ЧЛЕН-ФУНКЦИИ ЗА ВХОД/ИЗХОД

◉ istream

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

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

str - [НУЗ](#);

[size - 1](#) - максимален брой на символите, които ще бъдат извлечени;

delim - символ, който се нарича разделител.

Пример:

```
char str[10], str2[10];
```

```
std::cin.get(str, 10, '.').get(str2, 10, '?');
```

Резултат:

ПОТОК

```
abcd.defg.1234
```

```
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 - [НУЗ](#);

[size - 1](#) - максимален брой на символите, които ще бъдат извлечени;

delim - символ, който се нарича разделител.

Пример:

```
char str[10], str2[10];
```

```
std::cin.getline(str, 10, '.').getline(str2, 10, '.');
```

Резултат:

ПОТОК

abcd.defg.1234

str - abcd;

str2 - defg;

ЧЛЕН-ФУНКЦИИ ЗА ВХОД/ИЗХОД

◉ istream

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

arr - [масив](#);

[size](#) - максимален брой на символите, които ще бъдат извлечени;

Пример:

```
char arr[10];  
std::cin.read(arr, 10);
```

Резултат:

ПОТОК

0123456789ABCD

arr - 0123456789;

ЧЛЕН-ФУНКЦИИ ЗА ВХОД/ИЗХОД

◎ istream

`int` peek();

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

Пример:

```
std::cout << cin.peek();
```

Резултат:

ПОТОК

ABCD

65

ЧЛЕН-ФУНКЦИИ ЗА ВХОД/ИЗХОД

◉ istream

```
istream& putback(char ch);
```

Пример:

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

Резултат:

ПОТОК

ABCD

c1 == 'A', c2 == 'B'

59CD

59

СЪСТОЯНИЕ НА ПОТОК

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

`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.

Член-функции:

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

СЪСТОЯНИЕ НА ПОТОК

```
int main() {  
    // 0, 1, 2, 4  
    std::cout << std::cin.goodbit  
        << std::cin.eofbit  
        << std::cin.failbit  
        << std::cin.badbit  
        << std::endl;  
  
    double result;  
    std::cin >> result; // 'ABC3'  
    std::cout << std::cin.rdstate() << std::endl; // 2  
  
    std::cin.clear();  
    char symbol;  
    std::cin >> symbol;  
    std::cout << std::cin.rdstate() << std::endl;  
    std::cout << symbol << std::endl; // A  
  
    return 0;  
}
```

СЪСТОЯНИЕ НА ПОТОК

```
#include <iostream>
```

```
int main() {
```

```
    double x;
```

```
    std::cout << "x: "; // 3.5 || 'a'
```

```
    std::cin >> x;
```

```
    std::cout << x << std::endl;
```

```
    std::cout << "good(): " << std::cin.good() << std::endl
```

```
        << "bad(): " << std::cin.bad() << std::endl
```

```
        << "fail(): " << std::cin.fail() << std::endl
```

```
        << "eof(): " << std::cin.eof() << std::endl;
```

```
    return 0;
```

```
}
```

```
x: a
-9.25596e+061
good(): 0
bad(): 0
fail(): 1
eof(): 0
Press any key to continue . . .
```

СЪСТОЯНИЕ НА ПОТОК

```
// извличаме 3 символа
```

```
char ch1, ch2, ch3;
```

```
std::cin.get(ch1).get(ch2).get(ch3); //abc
```

```
std::cout << ch1 << ch2 << ch3 << std::endl; //abc
```

```
// опитваме се да върнем 4 символа
```

```
std::cin.putback('1').putback('2').putback('3').putback('k');
```

```
std::cout << std::cin.rdstate() << std::endl; // 0 || 4
```

```
std::cin.clear(); // възстановява cin, но има загуба на данни
```

```
// входни данни
```

```
std::cin.get(ch1).get(ch2).get(ch3); // 987
```

```
std::cout << ch1 << ch2 << ch3 << std::endl; // k98
```


ВРЕМЕ ЗА ВАШИТЕ ВЪПРОСИ