ПОТОК

гл.ас. д-р. Нора Ангелова

Входно-изходни операции

Вече познаваме такива ☺

Примери:

```
cout << ...;
cin >> ...;
```

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

ПОТОК

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

Представляват 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.

устройство  оперативна памет

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

устройство  оперативна памет

* *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

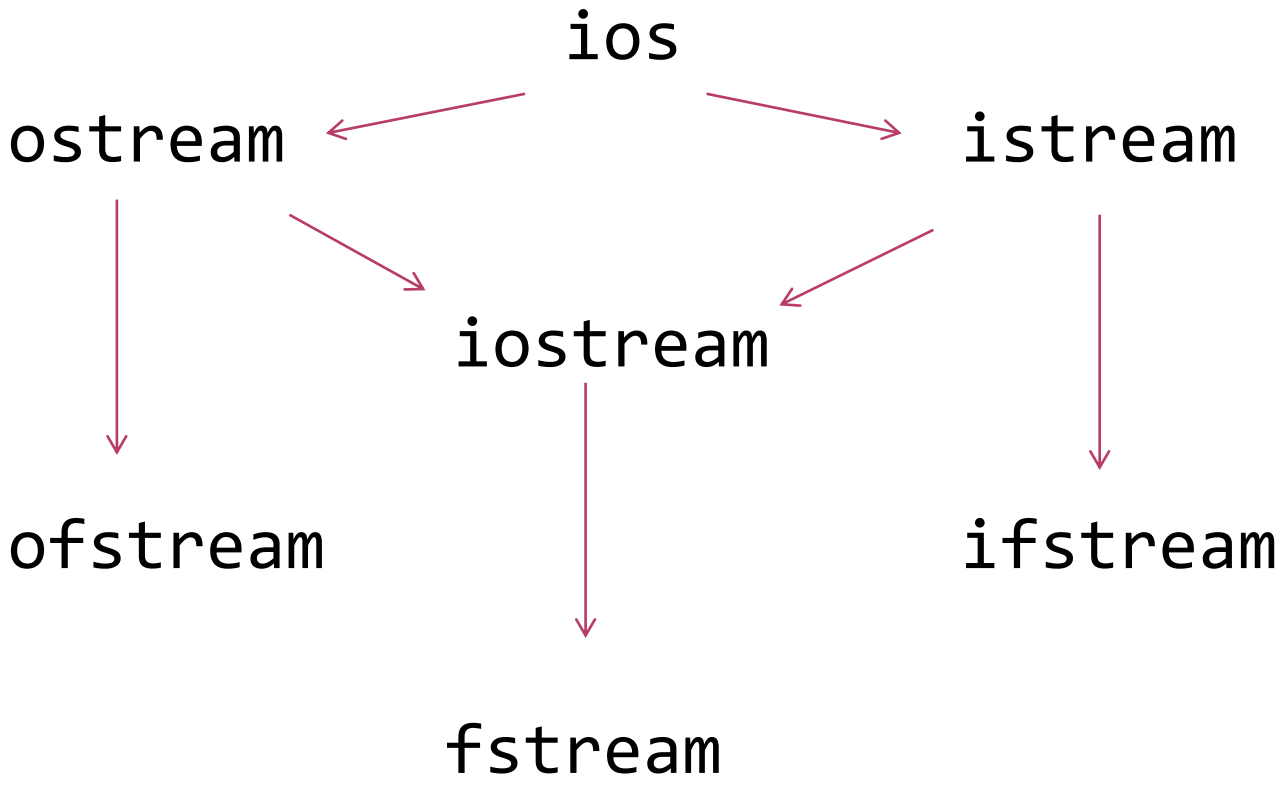
- **Изходен поток** – поток, който е свързан с определен приемник на данни.

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:

```
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:

```
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;
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];
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 - низ;
size - 1 - 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, '.');
```

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 - низ;
size - 1 - 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, '?');
```

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 - низ;
size - 1 - 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(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];
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:

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

Резултат:

поток

ABCD

65

Член-функции за вход/изход

- **istream**

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

Пример:

```
char c1, c2;
cin.get(c1).get(c2);
cin.putback('9').putback('5');
cin.get(c1).get(c2);
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 – iostate.

ios::goodbit – операциите са изпълнени успешно.
ios::eofbit – достигнат е край на файла (акитивира и failbit).
ios::failbit – последната входно/изходна операция е неуспешна.
ios::badbit – изпълнена е невалидна операция и има загубена информация.

Състояние на поток

```
int main() {
  // 0, 1, 2, 4
  cout << cin.goodbit << cin.eofbit << cin.failbit << cin.badbit << endl;

  double result;
  cin >> result; // 'ABC3'
  cout << cin.rdstate() << endl; // 2

  cin.clear();
  char c;
  cin >> c ;
  cout << cin.rdstate() << endl;
  cout << c << endl; // A
  system("pause");
  return 0;
}
```

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

```
#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;
}
```

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

// опитваме се да върнем 4 символа
cin.putback('1').putback('2').putback('3').putback('k');

cout << cin.rdstate() << endl; // 0 || 4
cin.clear(); // възстановява cin, но има загуба на данни

// входни данни
cin.get(ch1).get(ch2).get(ch3); // 987
cout << ch1 << ch2 << ch3 << endl; // k98
```

КРАЙ

?