

## IO类：不存在拷贝和赋值，只是引用

io类库类型和头文件有：

头文件	类型
iostream	istream, wistream, ostream, wostream, iostream, wiostream :从流中读取写入数据
fstream	ifstream, wifstream, ofstream, wofstream, fstream, wfstream: 从文件读入写入数据
sstream	istringstream, wistringstream, .....从string读取写入数据

io类的一些条件状态在ps279。

io输出缓冲相关知识：

```
1 cout << "hi" << endl;    // 换行并刷新缓冲区
2 cout << "hi" << flush;   // 刷新缓冲区，仅此而已
3 cout << "hi" << ends;    // 加个空字符并刷新缓冲区
4 cout << unitbuf;        // 此设计后，任何输出都立即刷新缓冲区
5 cout << nunitbuf;       // 恢复到默认状态
```

io关联输入和输出流：

当一个输入流被关联到一个输出流时，任何从输入流读取数据，都会刷新关联的输出流。只能关联输出流。

```
1 cin.tie(&cout); // cin关联cout,cout可以关联其他cout，关联同一个都可以，反正只能关联输出流 => 为了刷新
```

文件输入输出：

举个例子：

```
1 filename = "test";
2 outfilename = "out.txt";
3 ifstream in(filename); // 构造一个ifstream，并调用open方法打开此文件
4 ofstream out(outfilename);
5 in.open(filename + ".txt"); // 打开指定文件
6 bool inopenstate = in.is_open(); // 判断打开与否；可以直接放在if(in)里面进行判断。
7 然后 in,out跟cin、cout一样的操作，最后close关了就好了，不关在对象被销毁时会调用close()。
```

文件模式:

in	以读的方式打开
out	以写的方式打开
app	每次写操作前均定位到文件末尾
ate	打开文件后立即定位到文件末尾
trunc	截断文件
binary	以二进制方式进行io

```

1 ofstream out(filename,ofstream::out);
2 ofstream out1(filename,ofstream::out | ofstream::trunc);    //添加多个方式用“ | ”隔开
3 .....

```

**string流**：就像Java里面字符串缓冲一样#include <sstream>

```

1 sstream strm; // 定义一个未绑定的stringstream对象
2 sstream strm(s);    // .....,并保存一个string s的拷贝
3 strm.str(); // 返回strm所保存的string拷贝
4 strm.str(s);    // 将string s 拷贝到strm中

```

```

1 string line = "name 123";
2 istreamstringstream stringin(line);    // 输入
3 string name;
4 int num;
5 istreamstringstream >> name >> num;
6 ostreamstringstream stringout;    //输出
7 stringout << name << " " << age << flush;
8 cout << stringout.str() << endl;    //一块输出

```