**标准流**

cin，cout继承了标准流

int a;

// 如果这里输入字符，如输入 cc ，后续的cin将无法正常进行（注：如果这里输入的是 cc ，由于无法给 a，那么 cc 还在 cin 流中）

if (cin >> a)

{

/\* 如果流有效（成功输入） \*/

cout << a << endl;

}

cin >> a;

cout << cin.eof() << endl; // 如果流达到文件结束位置，返回 true（cin读取的不是文件）

cout << cin.bad() << endl; // 如果发生系统级错误，这里将返回 true

cout << cin.fail() << endl; // 如果流出错，这里将返回 true

cout << cin.good() << endl; // 如果流还可以读取，返回true

// 如果发生错误，但未发生系统级错误，这里将状态复位，这样，后续的 cin 又可以读取了

if (cin.fail() == true && cin.bad() == false)

{

cin.clear(); // 复位所有状态，

}

auto oldState = cin.rdstate(); // 读取cin状态

cin.setstate(oldState); // 设置cin状态

string s;

cin >> s;

// cout << s 不会立即输出，endl的作用是换行并刷新缓存区

cout << s << endl;

**文件流**

#include <fstream>

// 定义文件流对象

fstream f;

// 将文件流关联到main.d文件

f.open("./main.d");

// 取消文件流和main.d文件的管理

f.close();

// 文件输入流

ifstream ifs;

// 以读方式管理到main.d

ifs.open("./main.d", ifstream::in);

// 如果文件输入流有效

if (ifs)

{

string word;

// 读取输入流，>> 会以空格，作为分隔符

while (ifs >> word)

{

cout << word << endl;

}

ifs.close();

}

// 文件输出流

ofstream ofs;

// 以写，追加方式管理到文件new.txt

ofs.open("./new.txt", ofstream::out | ofstream ::app);

if (ofs)

{

// 写如流

ofs << "ABC";

ofs.close();

}

**String流**

istringstream iss;

iss.str("ab cd ef"); // 设置流字符

string word;

while (iss >> word)

{

cout << word << endl; // 依次输出 ab, cd, ef

}

ostringstream oss;

oss << "ab";

oss << "cd";

oss << "ef";

cout << oss.str() << endl; // str()获取流的字符，输出 abcdef