

# Lab2 问题 康艺潇 518431910002

```
int start, end;
char comma, type = ' ';
stringstream ss(cmd);
ss >> start;
if (ss.eof()) {
    cmdNull(start);
return; }
ss >> comma >> end >> type;
if (ss.good()) {
    if (type == 'n') {
        cmdNumber(start, end);
        return;
    } else if (type == 'd') {
        cmdDelete(start, end);
        return;
    }
}
```

请解释在Editor::dispatchCmd中下述代码的逻辑，尤其注意对stringstream的使用。

## 回答：

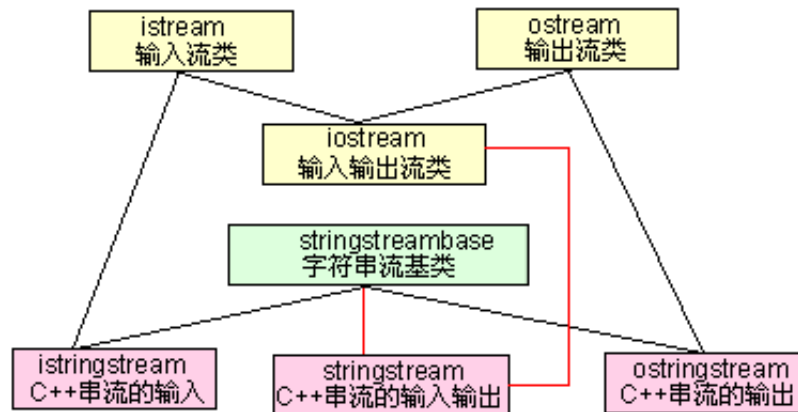
### 1. 代码逻辑：

1. 以cmd字符串构造stringstream ss用来进行字符串分割，不同类型的数据匹配（注1）。
2. 正常情况下进行到这一步，数据只可能为单个数字，【数字，数字n/d】。开头都是数字，所以讲第一个数字从ss中取出，输出到int型的start中。
3. 如果是单个数字，则调用eof（）函数判断是否遇到文件结束符时得到true。于是进入单行打印函数cmdNumber(start, end);完毕后return。
4. 如果不是单个数字，则把结束行数和类型输出到 end 和type里面。如果输出正常（注2），则进行类型匹配，如果是n则进入cmdNumber(start, end);函数完成后return。如果是d,则进入cmdDelete(start, end);函数完成后return。

### 2. stringstream使用说明

- 使用stringstream需要包括头文件#include

○ 继承关系



### 3. 注1：使用stringstream进行类型匹配。

```
#include <iostream>
#include <sstream>

using namespace std;

int main()
{
    stringstream str;
    str.str("1 56.7");
    //上述两个过程可以简单写成 stringstream str("1 56.7");
    cout << str.str() << endl;
    int a;
    float b;
    str >> a;
    cout << a << endl;
    str >> b;
    cout << b << endl;
    return 0;
}
```

上例中，构造字符串流的时候，空格会成为字符串参数的内部分界，例子中对a,b对象的输入"赋值"操作证明了这一点，字符串的空格成为了整型数据与浮点型数据的分解点，利用分界获取的方法我们事实上完成了字符串到整型对象与浮点型对象的拆分转换过程。

### 4. 注2：stream::good

Returns `true` 当没有任何 stream's *error state flags* (eofbit, failbit and badbit) 出现。

iostate value (member constant)	indicates
goodbit	No errors (zero value iostate)
eofbit	End-of-File reached on input operation
failbit	Logical error on i/o operation
badbit	Read/writing error on i/o operation

#举例

```
#include <iostream>
#include <string>
#include <sstream>
using namespace std;

int main()
{
    std::stringstream stream;
    string s = "ste";
    int i=0;
    stream<<s;
    if (stream.fail()) cout<<"fail!"<<endl;
}
```

```
fail!
Program ended with exit code: 0
```