

# C++ stringstream的妙用

转载 Lavi\_qq\_2910138025 2019-04-07 15:37:20 6591 收藏 9

分类专栏： 数据结构与算法 文章标签： C++ stringstream LeetCode

## 1 介绍

C++引入了ostringstream、istringstream、stringstream这三个类，要使用他们创建对象就必须包含sstream.h头文件。

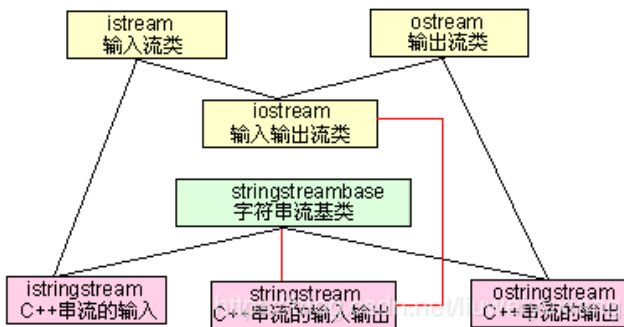
istringstream类用于执行C++风格的串流的输入操作。

ostringstream类用于执行C风格的串流的输出操作。

stringstream类同时可以支持C风格的串流的输入输出操作。

istringstream类是从istream和stringstreambase派生而来，ostringstream是从ostream和 stringstreambase派生而来，stringstream则是从iostream类和stringstreambase派生而来。

他们的继承关系如下图所示：



## 2 istringstream

istringstream是由一个string对象构造而来，istringstream类从一个string对象读取字符。

istringstream的构造函数原形如下：

```
1 | istringstream::istringstream(string str);
```

示例代码如下：

```
1 | #include <iostream>
2 | #include <sstream>
3 |
4 | using namespace std;
5 |
6 | int main()
7 | {
8 |     istringstream istr;
9 |     istr.str("1 56.7");
10 |    //上述两个过程可以简单写成 istringstream istr("1 56.7");
11 |    cout << istr.str() << endl;
12 |    int a;
13 |    float b;
```

```
14     istr >> a;
15     cout << a << endl;
16     istr >> b;
17     cout << b << endl;
18     return 0;
19 }
```

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。

str()成员函数的使用可以让istringstream对象返回一个string字符串（例如本例中的输出操作(cout<<istr.str();））。

## 3 ostringstream

ostringstream同样是由一个string对象构造而来，ostringstream类向一个string插入字符。

ostringstream的构造函数原形如下：

```
1 | ostringstream::ostringstream(string str);
```

示例代码如下：

```
1 | #include <iostream>
2 | #include <sstream>
3 | #include <string>
4 | #include<cstdlib>
5 | using namespace std;
6 | int main()
7 | {
8 |     ostringstream ostr;
9 |     //ostr.str("abc");
10 |    //如果构造的时候设置了字符串参数,那么增长操作的时候不会从结尾开始增加,而是修改原有数据,超出的部
11 |    ostr.put('d');
12 |    ostr.put('e');
13 |    ostr<<"fg";
14 |
15 |    string gstr = ostr.str();
16 |    cout<<gstr;
17 |    system("pause");
18 | }
```

在上例代码中，我们通过put()或者左移操作符可以不断向ostr插入单个字符或者是字符串，通过str()函数返回增长过后的完整字符串数据，但值得注意的一点是，当构造的时候对象内已经存在字符串数据的时候，那么增长操作的时候不会从结尾开始增加,而是修改原有数据,超出的部分增长。

[ basic\_stringbuf::str : Sets or gets the text in a string buffer without changing the write position. ]

## 4 stringstream

对于stringstream来说，不用我多说，大家也已经知道它是用于C++风格的字符串的输入输出的。

stringstream的构造函数原形如下：

```
1 | stringstream::stringstream(string str);
```

示例代码如下:

```
1 | #include <iostream>
2 | #include <sstream>
3 | #include <string>
4 | #include<cstdlib>
5 | using namespace std;
6 |
7 | int main()
8 | {
9 |     stringstream ostr("ccc");
10 |     ostr.put('d');
11 |     ostr.put('e');
12 |     ostr<<"fg";
13 |     string gstr = ostr.str();
14 |     cout<<gstr<<endl;
15 |
16 |     char a;
17 |     ostr>>a; //可以看到stringstream对象既可以做输入流也可以做输出流
18 |     cout<<a;
19 |     system("pause");
20 | }
```

## 4.1 stringstream用于数据类型转换

stringstream最常用于string与各种内置数据类型的转换。

示例代码如下：

```
1 | #include <iostream>
2 | #include <sstream>
3 | #include <string>
4 | #include<cstdlib>
5 | using namespace std;
6 |
7 | int main()
8 | {
9 |     stringstream sstr;
10 |     //-----int转string-----
11 |     int a=100;
12 |     string str;
13 |     sstr<<a;
14 |     sstr>>str;
15 |     cout<<str<<endl;
16 |     //-----string转char[]-----
17 |     sstr.clear();//如果你想通过使用同一stringstream对象实现多种类型的转换，请注意在每一次转换之后都必须
18 |     string name = "colinguan";
19 |     char cname[200];
20 |     sstr<<name;
21 |     sstr>>cname;
22 |     cout<<cname;
23 | }
```

```
24 |         system("pause");
    |     }
```

string转double/int:

```
1 | #include <iostream>
2 | #include <sstream>
3 |
4 | using namespace std;
5 |
6 | int main()
7 | {
8 |     double dVal;
9 |     int iVal;
10 |    string str;
11 |    stringstream ss;
12 |
13 |    // string -> double
14 |    str = "123.456789";
15 |    ss << str;
16 |    ss >> dVal;
17 |    cout << "dVal: " << dVal << endl;
18 |
19 |    // string -> int
20 |    str = "654321";
21 |    ss.clear();
22 |    ss << str;
23 |    ss >> iVal;
24 |    cout << "iVal: " << iVal << endl;
25 |
26 |    return 0;
27 | }
```

## 4.2 stringstream用于空格分割的字符串的切分

LeetCode : 557. 反转字符串中的单词 III

给定一个字符串，你需要反转字符串中每个单词的字符顺序，同时仍保留空格和单词的初始顺序。

示例 1:

```
1 | 输入: "Let's take LeetCode contest"
2 | 输出: "s'teL ekat edoCteeL tsetnoc"
```

注意：在字符串中，每个单词由单个空格分隔，并且字符串中不会有任何额外的空格。

```
1 | class Solution {
2 | public:
3 |     string reverseWords(string s) {
4 |         //使用C++的stringstream类和, string的reverse函数
5 |         if(s.size() == 0)
6 |             return "";
7 |         stringstream ss;
8 |     }
```

```
9      ss << s;
10     string p,str;
11     while(ss >> p){
12         reverse(p.begin(), p.end()); //string 可以调用reverse函数,string的表现像一个vector<char>
13         str = str + p + ' ';
14     }
15     string ans(str.begin(), str.begin()+str.size()-1);
16     return ans;
17 }
};
```

原文链接：