

C++语言基础

迂者 - 贺利坚

<http://blog.csdn.net/sxhelijian/>

<http://edu.csdn.net>





本节主题：

字符串流

文件流与字符串流

| | 文件流 | 字符串流 |
|------|----------------------------|---------------------------------------|
| 概 念 | 文件流是以外存文件为输入输出对象的数据流 | 字符串流也称为内存流，以内存中用户定义的字符数组(字符串)为输入输出的对象 |
| 相关流类 | ifstream, ofstream和fstream | stringstream |
| 头文件 | ifstream, ofstream和fstream | stringstream |

- 文件流类和字符串流类都是ostream,istream和iostream类的派生类，因此对它们的操作方法是基本相同的。

理解字符串流

❏ `float m,n; char op;`

❏ `cin>>m>>op>>n;`

输入是字符串形式，存储在输入缓冲区中

❏ 键盘输入: 234.5+23.6

❏ 结果

m ————— 234.5

op ————— '+'

n ————— 23.6

数据的内部表示：
二进制方式

输入输出缓冲区就是字符串流！

❏ `cout<<m<<' '<<n<<' '<<op;`

❏ 屏幕输出: 234.5 23.6 +

输出也是字符串形式，存储在输出缓冲区中

建立输出字符串流对象

- ❏ 字符串流类没有open成员函数，通过调用构造函数建立字符串流对象
- ❏ ostream类的构造函数的原型

```
ostream::ostream(char *buffer, int n, int mode=ios::out);           #include <ostream>
```

❏ buffer是指向字符数组首元素的指针，

❏ n为指定的流缓冲区的大小(一般选与字符数组的大小相同，也可以不同)

❏ mode指操作方式，默认为ios::out方式。

- ❏ 例：建立输出字符串流对象并与字符数组建立关联

```
char ch1[20];
```

```
ostream strout(ch1, 20);
```

建立输入字符串流对象

❏ `istream` 类的两个带参的构造函数，原型为：

`#include <sstream>`

`istream::istream(char *buffer);`

`istream::istream(char *buffer, int n);`

❏ `buffer`是指向字符数组首元素的指针，用它来初始化流对象(使流对象与字符数组建立关联)

❏ `n`是流缓冲区大小，可以只用字符数组中的一部分

❏ 例：建立输入字符串流对象

`char ch2[40];`

`istream strin(ch2);` //将字符数组ch2中的全部数据作为输入字符串流的内容

`istream strin(ch2,20);` //只将字符数组ch2中的前20个字符作为输入字符串流的内容

建立输入输出字符串流对象

❏ `stringstream`类提供的构造函数的原型为

`#include <stringstream>`

`stringstream::stringstream(char *buffer, int n, int mode);`

❏ `buffer`是指向字符数组首元素的指针，

❏ `n`为指定的流缓冲区的大小(一般选与字符数组的大小相同，也可以不同)

❏ `mode`指操作方式，默认为`ios::out`方式。

❏ 例

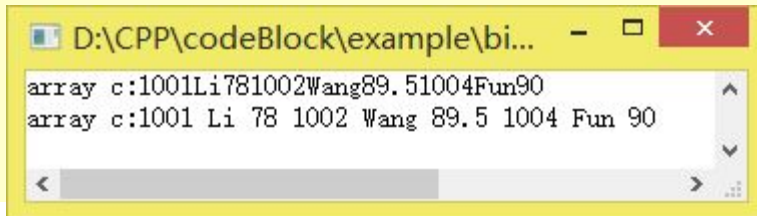
❏ `char ch3[80];`

❏ `stringstream strio(ch3, sizeof(ch3), ios::in | ios::out);`

例：“写”字符数组

```
struct student
{
    int num;
    char name[20];
    float score;
};
```

```
int main( )
{
    student stud[3]= {1001,"Li",78,1002,"Wang",89.5,1004,"Fun",90};
    char c[50];
    ostrstream strout1(c,30);
    for(int i=0; i<3; i++)
        strout1<<stud[i].num<<stud[i].name<<stud[i].score;
    strout1<<ends;    //ends是C++的I/O操作符，插入一个'\0'
    cout<<"array c:"<<c<<endl;
    ostrstream strout2(c,40);
    for(int i=0; i<3; i++)
        strout2<<stud[i].num<<" "<<stud[i].name<<" "<<stud[i].score<<" ";
    strout2<<ends;
    cout<<"array c:"<<c<<endl;
    return 0;
}
```

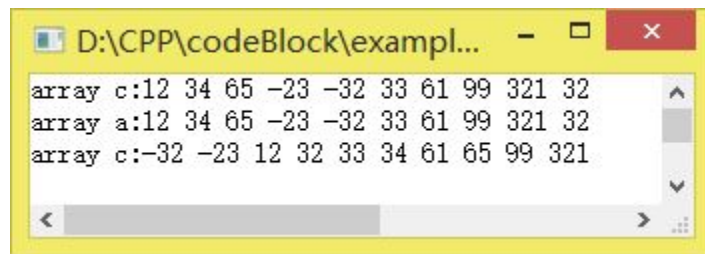


```
D:\CPP\codeBlock\example\bi...
array c:1001Li781002Wang89.51004Fun90
array c:1001 Li 78 1002 Wang 89.5 1004 Fun 90
```


以字符串流为中介交换数据

```
int main( )
{
    char c[50]="12 34 65 -23 -32 33 61 99 321 32";
    int a[10],i,j,t;
    cout<<"array c:"<<c<<endl;
    istrstream strin(c,sizeof(c));
    for(i=0; i<10; i++) strin>>a[i];
    cout<<"array a:";
    for(i=0; i<10; i++) cout<<a[i]<<" ";
    cout<<endl;
    //用起泡法对数组 a 排序 ... ..
    ostrstream strout(c,sizeof(c));
    for(i=0; i<10; i++) strout<<a[i]<<" ";
    strout<<ends;
    cout<<"array c:"<<c<<endl;
    return 0;
}
```

螺丝壳中做道场：以字符串流为中介，
ASCII、二进制形式互转。



```
D:\CPP\codeBlock\exampl...
array c:12 34 65 -23 -32 33 61 99 321 32
array a:12 34 65 -23 -32 33 61 99 321 32
array c:-32 -23 12 32 33 34 61 65 99 321
```

操作内存中的字符数组和文件的差异

- ❏ 输出时数据不是流向外存文件，而是流向内存中的一个存储空间；输入时从内存中的存储空间读取数据。
- ❏ 字符串流对象关联的不是文件，而是内存中的一个字符数组，因此不需要打开和关闭文件。
- ❏ 每个文件的最后都有一个文件结束符，表示文件的结束；而字符串流所关联的字符数组中没有相应的结束标志，用户要指定一个特殊字符`ends('\0')`作为结束符，在向字符数组写入全部数据后要写入此字符。

利用字符串流可以更灵活地操作数据

- ❑ 通过字符串流从字符数组读数据就如同从键盘读数据一样，可以从字符数组读入字符数据，也可以读入整数、浮点数或其他类型数据。
- ❑ 同一字符数组c可以先后与不同字符串流(strin或strout) 建立关联，分别对同一字符数组进行操作，甚至可以对字符数组交叉进行读写。
- ❑ 用输出字符串流向字符数组c写数据时，是从数组的首地址开始的，因此更新了数组的内容。
- ❑ 与字符串流关联的字符数组并不一定是专为字符串流而定义的数组，可以对该数组进行可施于一般字符数组的其他操作。
- ❑ 与字符串流关联的字符数组相当于内存中的临时仓库，可以以ASCII形式存放各种类型的数据，在需要时再从中读回来。其用法相当于标准设备(显示器与键盘)，同时具有诸多好处。

THANKS

本课程由 迂者-贺利坚 提供

CSDN网站：www.csdn.net
企业服务：<http://ems.csdn.net/>
人才服务：<http://job.csdn.net/>
CTO俱乐部：<http://cto.csdn.net/>
高校俱乐部：<http://student.csdn.net/>
程序员杂志：<http://programmer.csdn.net/>

CODE平台：<https://code.csdn.net/>
项目外包：<http://www.csto.com/>
CSDN博客：<http://blog.csdn.net/>
CSDN论坛：<http://bbs.csdn.net/>
CSDN下载：<http://download.csdn.net/>