

§ 13. 输入输出流

1850059

计1班

杨志远

13. 5. 字符串流

§ 13. 输入输出流

13. 5. 字符串流

13. 5. 1. 基本概念

§ 13. 输入输出流

13. 5. 字符串流

13. 5. 1. 基本概念

以内存中用户定义的string对象或流对象本身为输入/输出对象

§ 13. 输入输出流

13. 5. 字符串流

13. 5. 1. 基本概念

以内存中用户定义的string对象或流对象本身为输入/输出对象

★ 可以存放各种类型的数据

§ 13. 输入输出流

13. 5. 字符串流

13. 5. 1. 基本概念

以内存中用户定义的string对象或流对象本身为输入/输出对象

- ★ 可以存放各种类型的数据
- ★ 与标准输入输出流相同，进行ASCII码和二进制的相互转换

§ 13. 输入输出流

13. 5. 字符串流

13. 5. 1. 基本概念

以内存中用户定义的string对象或流对象本身为输入/输出对象

★ 可以存放各种类型的数据

★ 与标准输入输出流相同，进行ASCII码和二进制的相互转换

向字符数组存数据 \Leftrightarrow cout:

从字符数组取数据 \Leftrightarrow cin :

§ 13. 输入输出流

13. 5. 字符串流

13. 5. 1. 基本概念

以内存中用户定义的string对象或流对象本身为输入/输出对象

★ 可以存放各种类型的数据

★ 与标准输入输出流相同，进行ASCII码和二进制的相互转换

向字符数组存数据 \Leftrightarrow cout: 二进制 \Rightarrow ASCII

从字符数组取数据 \Leftrightarrow cin : ASCII \Rightarrow 二进制

§ 13. 输入输出流

13. 5. 字符串流

13. 5. 1. 基本概念

以内存中用户定义的string对象或流对象本身为输入/输出对象

- ★ 可以存放各种类型的数据
- ★ 与标准输入输出流相同，进行ASCII码和二进制的相互转换
 - 向字符数组存数据 \Leftrightarrow cout: 二进制 \Rightarrow ASCII
 - 从字符数组取数据 \Leftrightarrow cin : ASCII \Rightarrow 二进制
- ★ 不是文件，不需要打开和关闭

§ 13. 输入输出流

13. 5. 字符串流

13. 5. 2. 相关流对象的建立

§ 13. 输入输出流

13. 5. 字符串流

13. 5. 2. 相关流对象的建立

字符串输出流对象：

字符串输入流对象：

字符串输入/输出流对象：

§ 13. 输入输出流

13.5. 字符串流

13.5.2. 相关流对象的建立

字符串输出流对象:

`ostreamstream` 对象名(string对象名, 状态)

字符串输入流对象:

`istreamstream` 对象名(string对象名, 状态)

字符串输入/输出流对象:

`stringstream` 对象名(string对象名, 状态)

§ 13. 输入输出流

13.5. 字符串流

13.5.2. 相关流对象的建立

字符串输出流对象：

`ostreamstream` 对象名(string对象名, 状态)

字符串输入流对象：

`istreamstream` 对象名(string对象名, 状态)

字符串输入/输出流对象：

`stringstream` 对象名(string对象名, 状态)

★ 加 `#include <sstream>`

§ 13. 输入输出流

13. 5. 字符串流

13. 5. 3. 字符串流对象的使用

```
int main()  
{  
    ostreamstream ostream("Hello");  
    cout << ostream.str() << endl;  
    ostream << 123 << "abc";  
    cout << ostream.str() << endl << "*";  
}
```

观察cout的输出

§ 13. 输入输出流

13. 5. 字符串流

13. 5. 3. 字符串流对象的使用

```
int main()
{
    ostreamstream ostream("Hello");
    cout << ostream.str() << endl;
    ostream << 123 << "abc" << ends;
    cout << ostream.str() << endl << "*";
}
```

观察cout的输出

§ 13. 输入输出流

13. 5. 字符串流

13. 5. 3. 字符串流对象的使用

```
int main()
{
    ostringstream ostream("Hello");
    cout << ostream.str() << endl;
    ostream << 123 << "abc" << endl << ends;
    cout << ostream.str() << endl << "*";
}
```

观察cout的输出

§ 13. 输入输出流

13. 5. 字符串流

13. 5. 3. 字符串流对象的使用

★ 自动包含 ‘\0’ (ends), 不需要自行加入

§ 13. 输入输出流

13. 5. 字符串流

13. 5. 3. 字符串流对象的使用

```
int main()
{
    string str = "Hello";
    ostringstream ostream(str);
    cout << ostream.str() << endl;
    ostream << 123 << "abc" << endl << ends;
    cout << ostream.str() << endl << "*";
}
```

观察cout的输出

§ 13. 输入输出流

13. 5. 字符串流

13. 5. 3. 字符串流对象的使用

- ★ 自动包含 ‘\0’ (ends), 不需要自行加入
- ★ 定义ostream流对象时, 参数可省略
- ★ _Str参数为流对象中初始的字符串

§ 13. 输入输出流

13. 5. 字符串流

13. 5. 3. 字符串流对象的使用

```
int main()
{
    string str = "Hello";
    ostreamstream ostream(str, ios::out);
    cout << ostream.str() << endl;
    ostream << 123 << "abc" << endl << ends;
    cout << ostream.str() << endl << "*";
}
```

观察cout的输出

§ 13. 输入输出流

13. 5. 字符串流

13. 5. 3. 字符串流对象的使用

```
int main()
{
    string str = "Hello";
    ostringstream ostream(str, ios::app);
    cout << ostream.str() << endl;
    ostream << 123 << "abc" << endl << ends;
    cout << ostream.str() << endl << "*";
}
```

观察cout的输出

§ 13. 输入输出流

13. 5. 字符串流

13. 5. 3. 字符串流对象的使用

- ★ 自动包含 ‘\0’ (ends), 不需要自行加入
- ★ 定义ostream流对象时, 参数可省略
- ★ _Str参数为流对象中初始的字符串
- ★ 不同的_mode参数会产生不同的效果

§ 13. 输入输出流

13. 5. 字符串流

13. 5. 3. 字符串流对象的使用

```
int main()
{
    istringstream istream("Hello");
    int num; float fnum;
    cout << istream.str() << endl;
    istream.str("123 7.62");
    cout << istream.str() << endl;
    istream >> num;
    cout << num << endl;
    istream >> fnum;
    cout << fnum << endl;}
```

观察cout的输出

§ 13. 输入输出流

13. 5. 字符串流

13. 5. 3. 字符串流对象的使用

- ★ 自动包含 ‘\0’ (ends), 不需要自行加入
- ★ 定义 `istream` 流对象时, 参数可省略
- ★ `_Str` 参数为流对象中初始的字符串
- ★ 不同的 `_mode` 参数会产生不同的效果

§ 13. 输入输出流

13. 5. 字符串流

13. 5. 3. 字符串流对象的使用

观察cout的输出

```
int main()
{
    stringstream stream("Hello");
    int num; float fnum;
    cout << stream.str() << endl;
    stream.str("123 7.62 ");
    cout << stream.str() << endl;
    stream >> num; cout << num << endl;
    stream >> fnum; cout << fnum << endl;
    stream << 789 << "abc";
    cout << stream.str() << endl;}
```


§ 13. 输入输出流

13. 5. 字符串流

13. 5. 3. 字符串流对象的使用

观察cout的输出

```
int main()
{
    stringstream stream("Hello", ios::in);
    int num; float fnum;
    cout << stream.str() << endl;
    stream.str("123 7.62 ");
    cout << stream.str() << endl;
    stream >> num; cout << num << endl;
    stream >> fnum; cout << fnum << endl;
    stream << 789 << "abc";
    cout << stream.str() << endl;}
}
```

§ 13. 输入输出流

13. 5. 字符串流

13. 5. 3. 字符串流对象的使用

观察cout的输出

```
int main()
{
    stringstream stream("Hello", ios::out);
    int num; float fnum;
    cout << stream.str() << endl;
    stream.str("123 7.62 ");
    cout << stream.str() << endl;
    stream >> num; cout << num << endl;
    stream >> fnum; cout << fnum << endl;
    stream << 789 << "abc";
    cout << stream.str() << endl;}
}
```

§ 13. 输入输出流

13. 5. 字符串流

13. 5. 3. 字符串流对象的使用

- ★ 自动包含 ‘\0’ (ends), 不需要自行加入
- ★ 定义stringstream流对象时, 参数可省略
- ★ _Str参数为流对象中初始的字符串
- ★ 不同的_mode参数会产生不同的效果

P. 435-436 例13.14 (改)

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

int main()
{
    student stud[3] = { 1001, "Li", 78, 1002, "Wang", 89.5,
1004, "Fun", 90 };
    ostringstream strout;
    for (int i = 0; i < 3; i++)
        strout << stud[i].num << stud[i].name <<
stud[i].score << " ";
    cout << "array c:" << strout.str() << endl;
}
```

```
#include <iostream>
#include <sstream>
using namespace std;
int main()          //P.437 例13.15
{
    string str("12 34 65 -23 -32 33 61 99 321 32");
    int a[10], i, j, t;
    cout << "array c:" << str << endl;
    istringstream strin(str);
    for (i = 0; i < 10; i++)
        strin >> a[i]; //str中的内容逐个读入int a[10]中
    cout << "array a:";
    for (i = 0; i < 10; i++) //输出int a[10]的内容
        cout << a[i] << " ";
    cout << endl;
```

//进行排序

```
    for (i = 0; i < 9; i++)  
        for (j = 0; j < 9 - i; j++)  
            if (a[j] > a[j + 1]) {  
                t = a[j];  
                a[j] = a[j + 1];  
                a[j + 1] = t;  
            }  
}
```

//输出到str中（str刚才用做了输入流）

```
ostringstream strout;  
for (i = 0; i < 10; i++)  
    strout << a[i] << " ";  
cout << "array c:" << str << endl;  
}
```

§ 13. 输入输出流

13. 5. 字符串流

13. 5. 3. 字符串流对象的使用

与标准输入/输出流的区别：

★ 使用方法及转换方式相同

★ 字符串流对象可重复使用