IO库的一些特性

格式化输入与输出

每个iostream对象维护一个格式状态来控制IO如何格式化的细节。例如，格式状态控制，整型值是几进制？浮点值的精度？输入的宽度？等等。

操纵符（manipulator）可以修改留的格式状态，影响流的状态。

- 是一个函数或一个对象
- 可以用作输入或输出运算符的运算对象，例如endl
- 操纵符返回它处理的流的对象

endl就是一操纵符，输出一个换行符并刷新缓冲区。

操纵符有两大类控制：

- 控制数值输出形式
- 控制补白的数量和位置
- 改变格式状态的操纵符都是成对的，即设置/复原

> 当操纵符改变流的格式状态时，通常改变后的状态对所有后续IO都生效。

布尔值的输出格式

```
cout<<boolalpha;         // 修改输出bool值得格式
cout<<true<<" "<<false;// 打印true false
cout<<noboolalpha;       // 恢复默认的bool值输出格式
```

整型值的输出格式

默认是十进制。
三个操纵符，hex,oct,dec

```
cout<<oct<<20<<" "<<1024<<endl; // 输出24 2000
cout<<hex<<20<<" "<<1024<<endl; // 输出14 400
cout<<dec<<20<<" "<<1024<<endl; // 输出20 1024，默认输出格式
```

> hex,oct,dec 只影响整型数值，不影响浮点值

输出中指出进制

```
cout<<showbase;
cout<<oct<<20<<" "<<1024<<endl; // 输出024 2000，八进制前缀0
cout<<hex<<20<<" "<<1024<<endl; // 输出0x24 400，十六进制前缀0x
```

```
cout<<dec<<20<<" "<<1024<<endl; // 输出20 1024,默认输出格式
cout<<noshowcase;
```

控制浮点数格式

默认情况,

- 浮点值按六位数字精度打印;
- 如果浮点值没有小数部分,则不打印小数点;
- 根据浮点数的值选择打印成定点十进制或科学记数法形式
- 优先选择可读性好的格式:非常大或非常小的打印为科学记数法,其他为定点十进制

其余部分略。