输入输出流

C++的输入输出

输入输出的含义

程序的输入指的是从输入文件将数据传送给程序，程序的输出指的是从程序将数据传输给输出文件。C++的输入输出包括以下3方面的内容：

1. 对**系统指定的标准设备**的输入和输出。键盘输入、屏幕输出，简称**标准I/O**。
2. 以**外存磁盘文件**为对象进行输入输出，称为文件的输入输出，简称**文件I/O**。
3. 对**内存中指定的空间**进行输入和输出。通常指定一个字符串数组作为存储空间。简称**串I/O**。

C++的输入输出流

输入输出是数据传送的过程，数据如流水一样从一处向另一处。C++将此过程称为**流**。

在C++中，输入输出流被定义为类。C++的I/O库中的类称为**流类**。用流定义的对象称为**流对象**。

`cout` 和 `cin` 是 `iostream` 类的对象。

> iostream类库中有关的类

1. `istream`：通用输入流和其他输入流的基类
2. `ostream`：通用输出流和其他输出流的基类
3. `iostream`：通用输入输出流和其他输入输出流的基类

I/O类库中的常用流类

类名	作用	在哪个头文件中声明
ios	抽象基类	iostream
istream	通用输入流和其他输入流的基类	iostream
ostream	通用输出流和其他输出流的基类	iostream
iostream	通用输入输出流和其他输入输出流的基类	iostream
ifstream	输入文件流类	fstream
ofstream	输出文件流类	fstream
fstream	输入输出文件流类	fstream
istrstream	输入字符串流类	strstream
ostrstream	输出字符串流类	strstream
strstream	输入输出字符串流类	strstream

在C++中，流类库中定义了4个流对象：`cin`、`cout`、`cerr`和`clog`。

- `cin`：标准输入流对象，用于从键盘读取数据。
- `cout`：标准输出流对象，用于向屏幕输出数据。

- `cerr`：标准错误流对象，用于向屏幕输出出错信息。
- `clog`：标准日志流对象，用于向屏幕输出出错信息。

标准输出流

cout,cerr,clog

ostream类定义了3个输出流对象，即`cout`、`cerr`和`clog`。

1. **cout流对象**

    1. `cout`不是C++预定义的关键字，它是ostream流类的**对象**，在iostream头文件中定义。cout流中的数据是用流插入运算符`<<`顺序加入的。

    ```
    cout << "I" << "study C++" << "very hard.";
    ```

    按顺序将字符串"I"、"study C++"、"very hard."插入到cout流中，cout就将它们送到显示器。**cout流是容纳数据的载体，不是运算符。**

    2. 用`cout<<`输出基本类型的数据时，可以不考虑数据是什么类型，系统会判断数据的类型，并根据其类型选择调用与之匹配的**运算符重载**函数。
    3. `cout`流在内存中对应开辟了一个缓冲区，用来存放流中的数据，当向`cout`流插入一个`endl`时，无论缓冲区是否已满，都立即输出流中所有数据，然后插入一个换行符，并**刷新流**。如果插入一个换行符`\n`（如`cout<<a<<'\n';`），则只输出`a`和换行，而**不刷新**`cout`流。
    4. 在`iostream`中只对`<<`和`>>`运算符用于**标准类型数据**的输入输出进行了重载，但未对用户声明的类型数据的输入输出进行重载。如果用户声明了新的类型，并希望用`<<`和`>>`运算符对其进行输入输出，应对`<<`和`>>`运算符另作重载。

2. **cerr流对象**

`cerr`流对象是标准出错流。其作用是向标准出错设备输出有关出错信息。`cerr`与`cout`的作用和用法差不多。不同的是，`cout`流通常是传送到显示器输出，但也可以被**重定向**输出到磁盘文件，而`cerr`流中的信息只能在显示器输出。当调试程序时，不希望程序运行时的出错信息被送到其他文件，而要求在显示器上及时输出，这时应使用`cerr`。`cerr`流中的信息是用户指定的。
例子如下

```
cerr << "Error!" << endl;
```

3. **clog流对象**

    `clog`流对象也是标准出错流，它是“console log”的缩写。它的作用和`cerr`相同，都是在终端显示器上显示出错信息。

    它们之间只有一个区别：`cerr`是直接输出到显示器，不经过缓冲区，而`clog`的信息会先存放在缓冲区中，等缓冲区满了或者遇到`endl`时才输出到显示器。

标准类型数据的格式输出

有两种输入输出方式

1. 无格式输出
2. 有格式输出

输出数据时,有两种方法可以控制输出格式

1. 使用操纵符
2. 使用流对象的成员函数

注意,操纵符在头文件`iomanip`中声明

用流成员函数put输出字符

事实上,`put`与C语言中的`putchar`函数功能相同,都是向显示器输出一个字符.

标准输入流

cin流对象

与上一节类似,`cin`是`istream`类的对象,在`iostream`头文件中声明.]
也没啥好说的,就是输入

用流成员函数输入字符

get有三种形式:

1. 不带参数的get函数,其调用形式为`get(char &)`
2. 带参数的get函数,其调用形式为`get(void *,streamsize n,char delim)`
3. 重载的get函数,其调用形式为`get(istream &is,char *s,streamsize n)`

用成员函数`getline`输入字符串
其用法为`getline(istream &is,char *s,streamsize n)`,与重载的get函数类似,但`getline`函数读入的字符串中不包括`delim`字符.

istream类中的其他成员函数

1.`eof()`函数

表示输入流结束

2.`peek()`函数

作用是返回输入流中的下一个字符,但不从流中删除该字符.调用形式为`is.peek()`

3.`putback()`函数

作用是将一个字符插入到输入流中,作为下一个输入字符.调用形式为`is.putback(char c)`

4.`ignore()`函数

作用是跳过输入流中的指定个数字符,调用形式为`is.ignore(streamsize n,int delim)`

文件的输入输出

文件流类

以磁盘文件为对象的输入输出,必须定义一个文件流类对象,并指定文件名.对文件的输入输出是通过文件流类实现的.

字符串流

字符串流与文件流类似,但它是以字符串为对象的输入输出,他们有3点不同:

1. 输出时,字符串流中的数据并不写入磁盘文件,而是存入字符串流对象中.
2. 字符串流对象关联的不是磁盘文件,而是字符串.
3. 用户要自己指定一个特殊字符串作为字符串流的结束标志.

字符串流对象要通过构造函数来建立,其调用形式为`strstream str(char *s,int n,ios::openmode mode)`