流类库与输入/输出

北京邮电大学

网络与交换技术国家重点实验室

宽带网研究中心

I/O流的概念

- 当程序与外界环境进行信息交换时，存在着两个对象，一个是程序中的流对象，另一个是文件对象
- 流是一种抽象，它负责在数据的生产者和数据的消费者之间建立联系，并管理数据的流动
- 程序建立一个流对象，并指定这个流对象与某个文件对象建立连接，程序操作流对象，流对象通过文件系统对所连接的文件对象产生作用
- 读操作在流数据抽象中被称为（从流中）提取，写操作被称为（向流中）插入

输出流

- 重要的输出流类型（C++标准库中预定义）
  - ostream
  - ofstream

ostream

- 预先定义了如下的输出流对象：
  - cout 标准输出
  - cerr 标准错误输出，没有缓冲，发送给它的内容立即被输出。
  - clog 类似于cerr，但是有缓冲，缓冲区满时被输出。

插入运算符（<<）

- 插入 (<<) 运算符对所有标准C++数据类型预先进行了操作符的重载
- 用于将数据转换成输出字节串的形式，传送到一个输出流对象

控制输出格式

- 控制输出宽度
  - 为了调整输出，可以通过在流中放入setw操纵符或调用width成员函数为每个项指定输出宽度。
- 例： 使用 width 控制输出宽度

```
#include <iostream>
using namespace std;
int main()
{
  double values[] = {1.23,35.36,653.7,4358.24};
  for (int i=0;i<4;i++)
  {
    cout.width(10);
    cout << values[i] <<'\n';
  }
}
```

输出结果：

```
      1.23
     35.36
     653.7
   4358.24
```

例：使用*填充

```
#include <iostream>
using namespace std;
int main()
{
  double values[]={1.23,35.36,653.7,4358.24};
  for(int i=0; i<4; i++)
  {
     cout.width(10);
     cout.fill('*');
     cout<<values[i]<<'\n';
  }
}
```

输出结果：

```
******1.23
*****35.36
*****653.7
***4358.24
```

例：使用setw指定宽度

```
#include <iostream>
#include <iomanip>
using namespace std;
int main()
{
  double values[]={1.23,35.36,653.7,4358.24};
  char *names[]={"Zoot","Jimmy","Al","Stan"};
  for (int i=0;i<4;i++)
  {
      cout << setw(6)  << names[i];
      cout << setw(10) << values[i] << endl;
  }
}
```

输出结果：

```
  Zoot      1.23
 Jimmy     35.36
    Al     653.7
  Stan   4358.24
```

ofstream

- ofstream 类支持磁盘文件输出
- 如果在构造函数中指定一个文件名，当构造这个文件时该文件是自动打开的
  - ofstream myFile("filename",iosmode);
- 可以在调用默认构造函数之后使用open成员函数打开文件

```
ofstream myFile; //声明一个静态输出文件流对象
myFile.open("filename",iosmode);
    //打开文件，使流对象与文件建立联系
```

ofstream 成员函数

- open函数
  把流与一个特定的磁盘文件关联起来。
  需要指定打开模式。
- put函数
  把一个字符写到输出流中。
- write函数
  把内存中的一块内容写到一个输出文件流中
- seekp和tellp函数
  操作文件流的内部指针
- close函数
  关闭与一个输出文件流关联的磁盘文件
- 错误处理函数
  在写到一个流时进行错误处理
- << 插入操作符

例：文件输出

```
#include <fstream>
using namespace std;
struct Date
{
   int mo,da,yr;
};
int main()
{
    Date dt = {6,10,92};
    ofstream tfile("date.dat",ios::binary);
    tfile.write((char *) &dt,sizeof dt);
    tfile.close();

    ofstream txtfile(“Data.txt”);
    txtfile << dt.mo << dt.da << dt.yr << endl;
    txtfile.close();
}
```

输入流

- 重要的输入流类型：
  - istream类最适合用于顺序文本模式输入。**cin 是预定义的标准输入对象**
  - ifstream类支持磁盘文件输入
- 提取运算符 (>>)
  - 对于所有标准C++数据类型预定义了重载
  - 是从一个输入流对象获取数据最容易的方法

ifstream

- 如果在构造函数中指定一个文件名，在构造该对象时该文件便自动打开。

```
ifstream myFile("filename",iosmode);
```

- 在调用缺省构造函数之后使用open函数来打开文件。

```
ifstream myFile;  //建立一个文件流对象
        myFile.open("filename",iosmode);
                //打开文件"filename“
```

- 成员函数

open（）get（）getline（）read（）seekg（）tellg（）

close（）

例：文件输入

```
#include <iostream>
#include <fstream>
using namespace std;
int main()
{ char ch;
  ifstream tfile("payroll",ios::binary);
  if(tfile)
  { tfile.seekg(8);
    tfile.get(ch);
    while(tfile.good())
    {cout<<ch;  tfile.get(ch);
     }
   } else
     { cout<<"ERROR: Cannot open file 'payroll'."<<endl; }
  tfile.close();
}
```