6.文本文件的读写

下面以ifstream、ofstream为例说明。

由于ifstream、ofstream类继承自流类 istream 和 ostream，因此也可以使用常见的IO操作。

比如 >>、get、getline 常用于文本文件输入，而操作 <<、put 常用于输出。

【例6】用符号 << 和 put 函数向文本文件写入一些文字

```
#include <iostream>
#include <fstream>
using namespace std;
int main()
{     //打开文件
      ofstream  out("file.txt");
      if(!out) {
            cout<<"打开文件失败！"<<endl;
            return 1;
      }
      //写文件
      out<<"Welcome to ";
      char ch[]="Xi'an Jiaotong University.";
      int i=0;
      while(ch[i]!=0) {  out.put(ch[i]);  i++;  }
      out.close();    //关闭文件
      return 0;
 }
```

运行结束后，在工程目录下产生文件file.txt，内容如下：

Welcome to Xi'an Jiaotong University.

真正编程时只要用一条语句 **out<<ch** 即可

【例7】用符号 >> 和 get 函数读取文本文件

本例将上一个例子建立的文件内容读出并显示在屏幕上。

```
#include <iostream>
#include <fstream>
using namespace std;
int main()
{
        //打开文件
        ifstream  in("file.txt");
        if(!in)
        {
                cout<<"不可以打开文件"<<endl;
                return 1;
        }
```

```
    //读文件
    char ch[80];
    in>>ch;                    //读取第一个单词Welcome
    cout<<ch;
    in>>ch;                    //读取第二个单词to
    cout<<ch;
    while(in)                  //剩余部分用get函数读出并显示
    {
        char c=in.get();
        if(in)   cout<<c;
    }
    in.close();   //关闭文件
    return 0;
}
```

运行后，屏幕显示：

Welcome to Xi'an Jiaotong University.

如何判断读到达文件尾部

当没有读到文件尾部时，ifstream 对象 in 相当于一个true值，甚至直到读取完所有有效数据后，in仍然相当于true。这时再读取一次文件后，in才变成了相当于false值的对象。

```
……
while(in)
 {
        char c=in.get();
        if(in)     cout<<c;
 }
```

之所以要有 if 判断，是为了不输出最后一次读取的非正文数据

综合实例：统计平均成绩

假设一个文件file.txt，内容为学生成绩，每一行的形式为：姓名、数学成绩、英语成绩、物理成绩。数据中间用空格隔开。编写程序读取每一行的内容，计算出每人的平均成绩后，写入输出文件。输出文件名由用户输入，输出文件每一行的形式为：姓名、数学成绩、英语成绩、物理成绩、平均成绩。