



文件-综合例题

学生成绩统计管理系统

1. 题目内容与要求

- ① 有四个学生信息文本文件，内容分别如下：
 - 第一个文件是学号、姓名、班级等基本信息；
 - 第二个文件是学号和高数成绩；
 - 第三个文件是学号和大英成绩；
 - 第四个文件是学号和计算机成绩。
- ② 请将这些文件按学号匹配合并为一个新的文本文件，新文件的每行内容如下：

学生学号、姓名、班级、高数成绩、大英成绩、计算机成绩和平均成绩。

2. 学生信息类

```
class Student { // 学生信息类
public:
    int no;          // 学号
    string name;      // 姓名
    string classname; // 班级
    float math;       // 高数成绩
    float english;    // 大英成绩
    float computer;   // 计算机成绩
    float average;    // 平均成绩
};
```

3. 读取学生成绩文件函数(1)

```
float getscore(string filename,int stdno){  
    int stdno_temp;  
    float score;  
    ifstream in(filename.c_str());    // 打开成绩文件  
    if( !in ) { // 判断文件打开是否正确  
        cout<<"文件打开错误。"<<endl;  
        return 0;  
    }  
}
```

读取学生成绩文件函数(2)

```
bool flag=false;
while( in ) {
    in>>stdno_temp>>score;
    if(in) { // 读正确时
        if(stdno_temp==stdno) {
            flag=true;
            break;
        }
    }
}
if(!flag) {
    score=0;
}
in.close();
return score;
}
```

4. 学生信息写文件运算符重载函数

```
ostream & operator<<(ostream & out, Student & student)
{
    out<<student.no<<'\t'<<student.name
        <<'\t'<<student.classname;
    out<<'\t'<<student.math;
    out<<'\t'<<student.english;
    out<<'\t'<<student.computer;
    out<<'\t'<<student.average;
    out<<endl;
}
```

5. 算法分析

- ① 定义学号、姓名、班级、高数成绩、大英成绩、计算机成绩和平均成绩变量；
- ② 分别以读方式打开四个原始数据文件和以写方式打开一个目标文件；
- ③ 循环从四个原始数据文件中依次读取学号、姓名、班级、高数成绩、大英成绩和计算机成绩，并计算平均成绩。读取出错时转⑤；
- ④ 将这些信息写入目标文件中，转③；
- ⑤ 关闭所有文件。

6. 主函数框架程序代码

```
#include <iostream>
```

```
#include <fstream>
```

```
using namespace std;
```

```
<学生信息类的定义在此>
```

```
<学生成绩读取函数在此>
```

```
<学生信息输出运算符重载函数在此>
```

```
int main()
```

```
{
```

```
    <核心代码在此>
```

```
    return 1;
```

```
}
```

主函数核心程序代码（1）

```
// 打开文件
```

```
ifstream in("info.txt"); // 学生信息文件
```

```
ofstream out("student.txt"); // 学生完整信息文件
```

```
if( !in || !out ) { // 判断文件打开是否正确
```

```
    cout<<"文件打开错误。"<<endl;
```

```
    return 1;
```

```
}
```

主函数核心程序代码（2）

```
out<<"学号\t\t姓名\t\t班级\t\t高数\t\t大英\t\t计算机\t\t平均"<<endl;
while( in) {
    Student student;
    // 读文件
    in>>student.no>>student.name>>student.classname;
    if(!in) { // 读正确时才写文件
        break;
    }
    student.math=getscore("math.txt",student.no);
    student.english=getscore("english.txt",student.no);
    student.computer=getscore("computer.txt",student.no);
    student.average=(student.math+student.english+student.computer)/3;
    // 写文件
    out<<student;
}
```

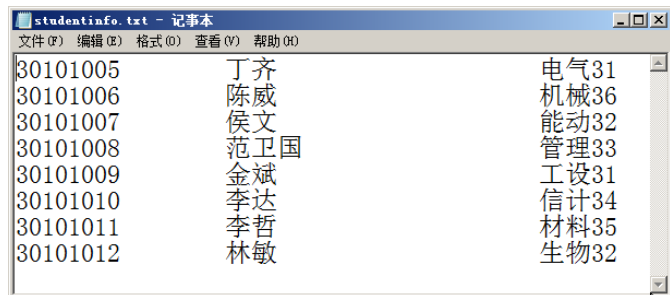
主函数核心程序代码（3）

```
//关闭文件  
out.close();  
in.close();
```

```
cout<<"student.txt文件建立成功，请查阅！"<<endl;
```

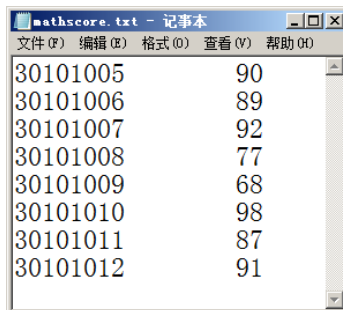


6. 运行结果



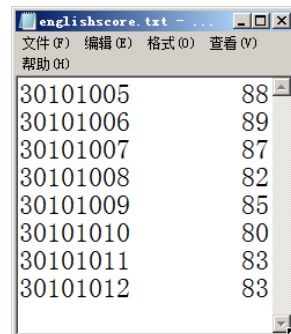
studentinfo.txt - 记事本

30101005	丁齐	电气31
30101006	陈威	机械36
30101007	侯文	能动32
30101008	范卫国	管理33
30101009	金斌	工设31
30101010	李达	信计34
30101011	李哲	材料35
30101012	林敏	生物32



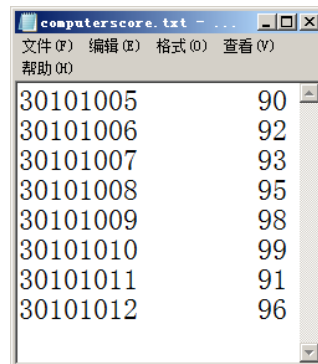
mathscore.txt - 记事本

30101005	90
30101006	89
30101007	92
30101008	77
30101009	68
30101010	98
30101011	87
30101012	91



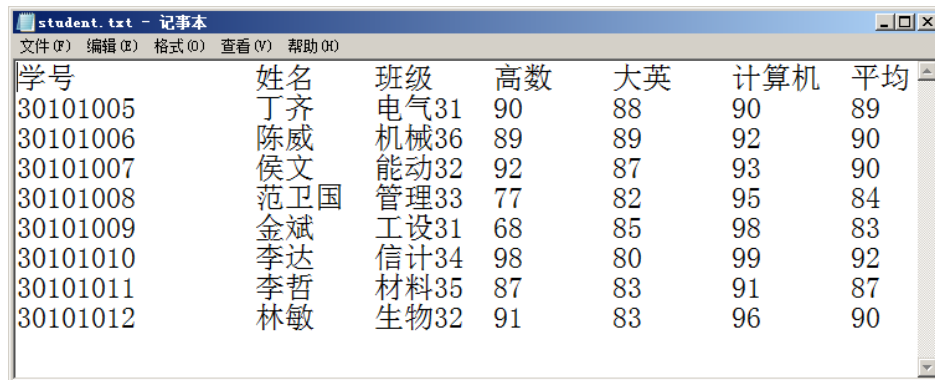
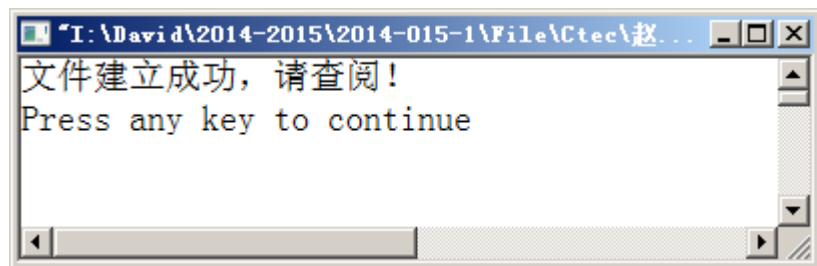
englishscore.txt - 记事本

30101005	88
30101006	89
30101007	87
30101008	82
30101009	85
30101010	80
30101011	83
30101012	83



computerscore.txt - 记事本

30101005	90
30101006	92
30101007	93
30101008	95
30101009	98
30101010	99
30101011	91
30101012	96



student.txt - 记事本

学号	姓名	班级	高数	大英	计算机	平均
30101005	丁齐	电气31	90	88	90	89
30101006	陈威	机械36	89	89	92	90
30101007	侯文	能动32	92	87	93	90
30101008	范卫国	管理33	77	82	95	84
30101009	金斌	工设31	68	85	98	83
30101010	李达	信计34	98	80	99	92
30101011	李哲	材料35	87	83	91	87
30101012	林敏	生物32	91	83	96	90

7. 延伸思考

问题1



如何对读文件进行>>运算符的重载？

问题2



怎样将最后的文件再行分割为多个文件？

8. 多文本文件输入输出主要步骤小结

- ✓ 包含文件流类头文件: `#include<fstream>`
- ✓ 打开文件读: `ifstream in1("a.txt");`
- ✓ 打开文件读: `ifstream in2("b.txt");`
- ✓ 打开文件读: `ifstream in3("c.txt");`
- ✓ 打开文件读: `ifstream in4("d.txt");`
- ✓ 打开文件写: `ofstream out1("e.txt");`
- ✓ 读文件: `in1 >> a;...`
- ✓ 写文件: `out1 << e;`
- ✓ 用完关闭文件流: `in1.close();.....`
`out1.close();`

例题的讲解就到这里！

谢谢！

