Assignment 12

191220022 丁一凡

一、 概念简答题

1. 分析说明C++语言的流类库中为什么要将ios类作为其派生类的虚基类

从流类库的基本结构可以看到， `ios` 类是 `istream` 类和 `ostream` 类的基类， 而 `iostream` 类通过多重继承 `istream` 类和 `ostream` 类而产生的。如果不将 `ios` 类作为其派生类的虚基类，可能会产生二义性。

2. 请简要概述文件缓冲区的作用，并结合其回答，程序中为什么要显示的关闭文件

文件缓冲区暂时存放读写期间的文件数据，减少读取外部存储器的次数

关闭文件才能把暂存在内存缓冲区中的内容写入文件中，如果不显示关闭文件，可能会导致缓冲区中的数据丢失或者无意中修改

二、 代码编程题

1.

```
istream &operator >> (istream &input, Complex &C)
{
    input >> C.real >> C.imag;
    return input;
}

ostream &operator << (ostream &output, const Complex &C)
{
    output << C.real;
    if(C.imag > 0)
    {
        output << "+" << C.imag << "i";
    }
    else if(C.imag < 0)
    {
        output << C.imag << "i";
    }
    return output;
}
```

```
template <class T>
istream &operator >> (istream &input, Matrix<T> &M)
{
    int col, row;
    input >> row >> col;
    M = Matrix<Complex> (row, col);
    for(int i = 0; i < row; i++)
    {
        for(int j = 0; j < col; j++)
        {
            input >> M[i][j].real >> M[i][j].imag;
        }
    }
    return input;
}

template <class T>
ostream &operator << (ostream &output, const Matrix<T> &M)
{
    output << M.getRow() << " " << M.getCol() << endl;
    for(int i = 0; i < M.getRow(); i++)
    {
        for(int j = 0; j < M.getCol(); j++)
        {
            output << M[i][j];
        }
        output << endl;
    }
    return output;
}
```

2.

```
int main()
{
    fstream io_file("test.txt", ios::in|ios::out);
    io_file.seekg(0);
    for(int i = 0; i <= 100; i++)
    {
        io_file << i;
    }
    io_file.close();

    io_file.open("test.txt", ios::in|ios::out);
    string t;
    for(int i = 0; i < 10; i++)
    {
        io_file.seekg(9 + i * 20);
        io_file.read(t,2);
```

```
        cout << t << endl;
        t.clear();
    }
    io_file.seekg(189);
    io_file.read(t,3);
    cout << t << endl;
    io_file.close();
    return 0;
}
```

3.

```
enum Sex {MALE, FEMALE};

struct Student{
    string id, name;
    Sex sex;
    double score;
}

bool compare(const Student& s1, const Student& s2)
{
    return s1.score > s2.score;
}

class ScoreList
{
    vector<Student> list;
public:
    ScoreList() {;}
    void KeyboardIn()
    {
        int num;
        cin >> num;
        Student stu;
        string sex;
        for(int i = 0; i < num; i++)
        {
            cin >> stu.id >> stu.name >> sex >> stu.score;
            if(sex == "MALE")
                stu.sex = MALE;
            else
                stu.sex = FEMALE;
            list.push_back(stu);
        }

        WriteFile();
    }
    void ReadFile()
    {
        ifstream in_file("a.txt", ios::in);
        if(!in_file)
            exit(-1);
```

```
    Student stu;
    string sex;
    while(!in_file.eof())
    {
        in_file >> stu.id >> stu.name >> sex >> stu.score;
        if(sex == "MALE")
            stu.sex = MALE;
        else
            stu.sex = FEMALE;
        list.push_back(stu);
    }
    in_file.close();
}
void Top3()
{
    sort(list.begin(), list.end(), compare);
    ofstream out_file("b.txt", ios::out);
    if(!out_file)
        exit(-1);
    for(int i = 0; i < min(list.size(), 3); i++)
    {
        out_file << list[i].id << " " << list[i].name << " ";
        if(list[i].sex == MALE)
            out_file << "MALE" << " ";
        else
            out_file << "FEMALE" << " ";
        out_file << list[i].score << endl;
    }
    out_file.close();
}
void Average()
{
    double boy_score = 0, girl_score = 0, boy_num = 0, girl_num = 0;
    for(int i = 0; i < list.size(); i++)
    {
        if(list[i].sex == MALE)
        {
            boy_score += list[i].score;
            boy_num++;
        }
        else
        {
            girl_score += list[i].score;
            girl_num++;
        }
    }
    double boy_ave = boy_score / boy_num;
    double girl_ave = girl_score / girl_num;

    ofstream out_file("c.txt", ios::out);

    //string sex;
    for(int i = 0; i< list.size(); i++)
    {
        if(list[i].sex == MALE)
        {
            if(list[i].score < boy_ave)
            {
```

```
                    out_file << list[i].id << " " << list[i].name << " MALE " <<
list[i].score << endl;
                }
            }
            else
            {
                if(list[i].score < girl_ave)
                {
                    out_file << list[i].id << " " << list[i].name << " FEMALE "
<< list[i].score << endl;
                }
            }
        }
        out_file.close();
    }
    void Add()
    {
        int num;
        cin >> num;
        Student stu;
        string sex;
        for(int i = 0; i < num; i++)
        {
            cin >> stu.id >> stu.name >> sex >> stu.score;
            if(sex == "MALE")
                stu.sex = MALE;
            else
                stu.sex = FEMALE;
            stu.score = stu.score * 0.9;
            list.push_back(stu);
        }

        WriteFile();
    }
    void WriteFile()
    {
                ofstream out_file("a.txt", ios::out);
        if(!out_file)
            exit(-1);
        for(int i = 0; i < list.size(); i++)
        {
            out_file << list[i].id << " " << list[i].name << " ";
            if(list[i].sex == MALE)
                out_file << "MALE" << " ";
            else
                out_file << "FEMALE" << " ";
            out_file << list[i].score << endl;
        }
        out_file.close();
    }
};
```