第七次作业

第一题

```cpp
#include <iostream>
using namespace std;
class People {
public:

    class Date {
    public:
        Date(int y,int m,int d):year(y),month(m),day(d) {
        }

        ~Date() {
        }
        int year, month, day;
        friend ostream& operator <<(ostream& o, Date& d);
    private:

    };

    char name[11];
    char number[7];
    char sex[3];
    char id[16];
    Date birthday;
    People(const char name[11], const char number[7], const char sex[3],const char id[16],int y,int m,int d):birthday(y,m,d) {
        strcpy_s(this->name, 11,name);
        strcpy_s(this->number, 7,number);
        strcpy_s(this->sex, 3,sex);
        strcpy_s(this->id, 16,id);
    }
```

```
30
31      ~People() {
32      }
33      People(const People& r) :birthday(r.birthday) {
34          strcpy_s(this->name, 11,r.name);
35          strcpy_s(this->number,7, r.number);
36          strcpy_s(this->sex,3, r.sex);
37          strcpy_s(this->id,16, r.id);
38      }
39      inline void show() {
40          cout << "name: " << name << "   " << "number: " << number
    << "   " << "sex: " << sex << "   " << "id:   " << id << "   " <<
    "birthday:  "<<birthday;
41      }
42  private:
43
44  };
45
46  int main()
47  {
48      People p("alice", "1", "wo", "12345678",2003, 1, 1);
49      People p1(p);
50      p1.show();
51  }
52
53  ostream& operator<<(ostream& o, People::Date& d) {
54      o << d.year << "年 " << d.month << "月 " << d.day << "日 ";
55      return o;
56  }
57
```

```
name: alice  number: 1  sex: wo  id:  12345678  birthday:  2003年 1月 1日
```

第二、三题

```
1  #include <iostream>
2  using namespace std;
3  class People {
```

```
public:
    class Date {
    public:
        Date(int y, int m, int d) :year(y), month(m), day(d) {
        }

        ~Date() {
        }
        int year, month, day;
        friend ostream& operator <<(ostream& o, Date& d);
    private:

    };

    People(const char name[11], const char number[7], const char sex[3], const char id[16], int y, int m, int d) :birthday(y, m, d) {
        strcpy_s(this->name, 11, name);
        strcpy_s(this->number, 7, number);
        strcpy_s(this->sex, 3, sex);
        strcpy_s(this->id, 16, id);
        cout << "People constructed"<<endl;
    }

    ~People() {
        cout << "people desturcted" << endl;
    }
    People(const People& r) :birthday(r.birthday) {
        strcpy_s(this->name, 11, r.name);
        strcpy_s(this->number, 7, r.number);
        strcpy_s(this->sex, 3, r.sex);
        strcpy_s(this->id, 16, r.id);
    }
    inline void show() {
        cout << "name: " << name << "  " << "number: " << number << "  " << "sex: " << sex << "  " << "id:  " << id << "  " << "birthday:  " << birthday;
    }

    char name[11];
    char number[7];
    char sex[3];
    char id[16];
```

```
    Date birthday;
    void eject() {
        this->~People();
    }
private:

};
class teacher :virtual public People{
public:
    teacher(const char name[11], const char number[7], const char sex[3], const char id[16], int y, int m, int d, const char ps[11],const char de[21]):People(name,number,sex,id,y,m,d) {
        strcpy_s(this->principalship, 11, ps);
        strcpy_s(this->department, 21, de);
        cout << "teacher constructed"<<endl;
    }
    teacher(const teacher& r) :People(r) {
        strcpy_s(this->principalship, 11, r.principalship);
        strcpy_s(this->department, 21, r.department);
    }
    ~teacher() {
        cout << "teacher desturcted" << endl;
    }
    char principalship[11];
    char department[21];
    inline void show() {
        People::show();
        cout <<"principalship: "<< principalship << "  " <<"department: "<< department << "  ";
    }void eject() {
        this->~teacher();
    }
private:

};

class student :virtual public People {
public:
    student(const char name[11], const char number[7], const char sex[3], const char id[16], int y, int m, int d,const char cn[7]):People(name,number,sex,id,y,m,d) {
        strcpy_s(this->classNo, 7, cn);
        cout << "student constructed" << endl;
```

```
    }
    student(const student& r):People(r) {
        strcpy_s(this->classNo, 7, r.classNo);
    }
    ~student() {
        cout << "student desturcted" << endl;
    }

    char classNo[7];
    inline void show() {
        People::show();
        cout << "classNo: " << classNo << "   ";
    }
    void eject() {
        this->~student();
    }
private:

};
class graduate :public student{
public:
    graduate(const char name[11], const char number[7], const char sex[3], const char id[16], int y, int m, int d, const char cn[7],const char su[21],const char te[30]):People(name, number, sex, id, y, m, d),student(name,number,sex,id,y,m,d,cn) {
        strcpy_s(this->subject, 21, su);
        strcpy_s(this->advisor, 30, te);
        cout << "graduate constructed" << endl;
    }

    graduate(const graduate& r):student(r),People(r) {
        strcpy_s(this->subject, 21, r.subject);
        strcpy_s(this->advisor, 30, r.advisor);
    }
    ~graduate() {
        cout << "graduate desturcted" << endl;
    }
    char subject[21];
    char advisor[30];
    inline void show() {
        student::show();
        cout << "subject: " << subject << "   " << "advisor: " << advisor << "   ";
```

```
    }
    void eject() {
        this->~graduate();
    }
private:

};
class TA :public teacher,public graduate {
public:
    TA(const char name[11], const char number[7], const char sex[3], const char id[16], int y, int m, int d, const char cn[7], const char su[21], const char te[30], const char ps[11], const char de[21]):People(name,number,sex,id,y,m,d),teacher(name,number,sex,id,y,m,d,ps,de),graduate(name,number,sex,id,y,m,d,cn,su,te) {
        cout << "TA constructed" << endl;
    }
    TA(const TA& r):People(r),teacher(r),graduate(r) {

    }
    ~TA() {
        cout << "TA desturcted" << endl;
    }
    inline void show() {
        graduate::show();
        cout << "principalship: " << principalship << "  " << "department: " << department << "  ";
    }
    void eject() {
        this->~TA();
    }
private:

};

int main() {
    TA ta("alice", "1", "wo", "12345678", 2003, 1, 1,"123456","electronic","yang fang","ta","department of ee");
    ta.show();


}

```

```
ostream& operator<<(ostream& o, People::Date& d) {
    o << d.year << "年 " << d.month << "月 " << d.day << "日 ";
    return o;
}

```

```
People constructed
teacher constructed
student constructed
graduate constructed
TA constructed
name: alice  number: 1  sex: wo  id:  12345678  birthday:  2003年 1月 1日 classNo: 123456  subject: electronic  advisor: yang fang  principalship: ta  depa
rtment: department of ee  TA desturcted
graduate desturcted
student desturcted
teacher desturcted
people desturcted
```

不使用虚继承则People会被构造两遍，同时graduate和TA类无需实现People构造函数

```
#include <iostream>
using namespace std;
class People {
public:
    class Date {
    public:
        Date(int y, int m, int d) :year(y), month(m), day(d) {}
        ~Date() {}
        int year, month, day;
        friend ostream& operator <<(ostream& o, Date& d);
    private:
    };
    People(const char name[11], const char number[7], const char sex[3], const char id[16], int y, int m, int d) :birthday(y, m, d) {
        strcpy_s(this->name, 11, name);
```

```
        strcpy_s(this->number, 7, number);
        strcpy_s(this->sex, 3, sex);
        strcpy_s(this->id, 16, id);
        cout << "People constructed" << endl;
    }
    ~People() {cout << "people desturcted" << endl;}
    People(const People& r) :birthday(r.birthday) {
        strcpy_s(this->name, 11, r.name);
        strcpy_s(this->number, 7, r.number);
        strcpy_s(this->sex, 3, r.sex);
        strcpy_s(this->id, 16, r.id);
    }
    inline void show() {
        cout << "name: " << name << "  " << "number: " << number
<< "  " << "sex: " << sex << "  " << "id:  " << id << "  " <<
"birthday:  " << birthday;
    }
    char name[11];
    char number[7];
    char sex[3];
    char id[16];
    Date birthday;
    void eject() {this->~People();}
private:
};
class teacher :public People {
public:
    teacher(const char name[11], const char number[7], const char
sex[3], const char id[16], int y, int m, int d, const char
ps[11], const char de[21]) :People(name, number, sex, id, y, m,
d) {
        strcpy_s(this->principalship, 11, ps);
        strcpy_s(this->department, 21, de);
        cout << "teacher constructed" << endl;
    }
    teacher(const teacher& r) :People(r) {
        strcpy_s(this->principalship, 11, r.principalship);
        strcpy_s(this->department, 21, r.department);
    }
    ~teacher() {cout << "teacher desturcted" << endl;}
    char principalship[11];
    char department[21];
    inline void show() {
```

```
53          People::show();
54          cout << "principalship: " << principalship << "  " <<
"department: " << department << "  ";
55      }
56      void eject() {this->~teacher();}
57  private:
58  };
59  class student :public People {
60  public:
61      student(const char name[11], const char number[7], const char
sex[3], const char id[16], int y, int m, int d, const char cn[7])
:People(name, number, sex, id, y, m, d) {
62          strcpy_s(this->classNo, 7, cn);
63          cout << "student constructed" << endl;
64      }
65      student(const student& r) :People(r) {
66          strcpy_s(this->classNo, 7, r.classNo);
67      }
68      ~student() {cout << "student desturcted" << endl;}
69      char classNo[7];
70      inline void show() {
71          People::show();
72          cout << "classNo: " << classNo << "  ";
73      }
74      void eject() {this->~student();}
75  private:
76  };
77  class graduate :public student {
78  public:
79      graduate(const char name[11], const char number[7], const
char sex[3], const char id[16], int y, int m, int d, const char
cn[7], const char su[21], const char te[30]) :student(name,
number, sex, id, y, m, d, cn) {
80          strcpy_s(this->subject, 21, su);
81          strcpy_s(this->advisor, 30, te);
82          cout << "graduate constructed" << endl;
83      }
84      graduate(const graduate& r) :student(r){
85          strcpy_s(this->subject, 21, r.subject);
86          strcpy_s(this->advisor, 30, r.advisor);
87      }
88      ~graduate() {cout << "graduate desturcted" << endl;}
89      char subject[21];
```

```
90      char advisor[30];
91      inline void show() {
92          student::show();
93          cout << "subject: " << subject << "  " << "advisor: " <<
    advisor << "  ";
94      }
95      void eject() {this->~graduate();}
96  private:
97  };
98  class TA :public teacher, public graduate {
99  public:
100     TA(const char name[11], const char number[7], const char
    sex[3], const char id[16], int y, int m, int d, const char cn[7],
    const char su[21], const char te[30], const char ps[11], const
    char de[21]) :teacher(name, number, sex, id, y, m, d, ps, de),
    graduate(name, number, sex, id, y, m, d, cn, su, te) {
101         cout << "TA constructed" << endl;
102     }
103     TA(const TA& r) :teacher(r), graduate(r) {}
104     ~TA() {cout << "TA desturcted" << endl;}
105     inline void show() {
106         graduate::show();
107         cout << "principalship: " << principalship << "  " <<
    "department: " << department << "  ";
108     }
109     void eject() {this->~TA();}
110 private:
111 };
112 int main() {
113     TA ta("alice", "1", "wo", "12345678", 2003, 1, 1, "123456",
    "electronic", "yang fang", "ta", "department of ee");
114     ta.show();
115 }
116 ostream& operator<<(ostream& o, People::Date& d) {
117     o << d.year << "年 " << d.month << "月 " << d.day << "日 ";
118     return o;
119 }
120
```

```
People constructed
teacher constructed
People constructed
student constructed
graduate constructed
TA constructed
name: alice  number: 1  sex: wo  id:  12345678  birthday:  2003年 1月 1日 classNo: 123456  subject: electronic  advisor:
 yang fang  principalship: ta  department: department of ee  TA desturcted
graduate desturcted
student desturcted
people desturcted
teacher desturcted
people desturcted
```

第四题

构造：继承的B1(1) -> memberB1(2) -> j(3) -> memberB2(0)

析构： memberB2 -> j -> memberB1 -> 继承的B1s