

第七次作业

第一题

```
1  #include <iostream>
2  using namespace std;
3  class People {
4  public:
5
6      class Date {
7      public:
8          Date(int y,int m,int d):year(y),month(m),day(d) {
9              }
10
11          ~Date() {
12              }
13          int year, month, day;
14          friend ostream& operator <<(ostream& o, Date& d);
15      private:
16
17      };
18
19      char name[11];
20      char number[7];
21      char sex[3];
22      char id[16];
23      Date birthday;
24      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) {
25          strcpy_s(this->name, 11,name);
26          strcpy_s(this->number, 7,number);
27          strcpy_s(this->sex, 3,sex);
28          strcpy_s(this->id, 16,id);
29      }
```

```

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 {

```

```

4 public:
5     class Date {
6     public:
7         Date(int y, int m, int d) :year(y), month(m), day(d) {
8         }
9
10        ~Date() {
11        }
12        int year, month, day;
13        friend ostream& operator <<(ostream& o, Date& d);
14    private:
15
16    };
17
18    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) {
19        strcpy_s(this->name, 11, name);
20        strcpy_s(this->number, 7, number);
21        strcpy_s(this->sex, 3, sex);
22        strcpy_s(this->id, 16, id);
23        cout << "People constructed"<<endl;
24    }
25
26    ~People() {
27        cout << "people desturcted" << endl;
28    }
29    People(const People& r) :birthday(r.birthday) {
30        strcpy_s(this->name, 11, r.name);
31        strcpy_s(this->number, 7, r.number);
32        strcpy_s(this->sex, 3, r.sex);
33        strcpy_s(this->id, 16, r.id);
34    }
35    inline void show() {
36        cout << "name: " << name << " " << "number: " << number
<< " " << "sex: " << sex << " " << "id: " << id << " " <<
"birthday: " << birthday;
37    }
38
39    char name[11];
40    char number[7];
41    char sex[3];
42    char id[16];

```

```

43     Date birthday;
44     void eject() {
45         this->~People();
46     }
47 private:
48
49 };
50 class teacher :virtual public People{
51 public:
52     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) {
53         strcpy_s(this->principalship, 11, ps);
54         strcpy_s(this->department, 21, de);
55         cout << "teacher constructed"<<endl;
56     }
57     teacher(const teacher& r) :People(r) {
58         strcpy_s(this->principalship, 11, r.principalship);
59         strcpy_s(this->department, 21, r.department);
60     }
61     ~teacher() {
62         cout << "teacher destrcted" << endl;
63     }
64     char principalship[11];
65     char department[21];
66     inline void show() {
67         People::show();
68         cout <<"principalship: "<< principalship << " "
<<"department: "<< department << " ";
69     }void eject() {
70         this->~teacher();
71     }
72 private:
73
74 };
75
76 class student :virtual public People {
77 public:
78     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) {
79         strcpy_s(this->classNo, 7, cn);
80         cout << "student constructed" << endl;

```

```

81     }
82     student(const student& r):People(r) {
83         strcpy_s(this->classNo, 7, r.classNo);
84     }
85     ~student() {
86         cout << "student destrcted" << endl;
87     }
88
89     char classNo[7];
90     inline void show() {
91         People::show();
92         cout << "classNo: " << classNo << " ";
93     }
94     void eject() {
95         this->~student();
96     }
97 private:
98
99 };
100 class graduate :public student{
101 public:
102     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) {
103         strcpy_s(this->subject, 21, su);
104         strcpy_s(this->advisor, 30, te);
105         cout << "graduate constructed" << endl;
106     }
107
108     graduate(const graduate& r):student(r),People(r) {
109         strcpy_s(this->subject, 21, r.subject);
110         strcpy_s(this->advisor, 30, r.advisor);
111     }
112     ~graduate() {
113         cout << "graduate destrcted" << endl;
114     }
115     char subject[21];
116     char advisor[30];
117     inline void show() {
118         student::show();
119         cout << "subject: " << subject << " " << "advisor: " <<
advisor << " ";

```

```

120     }
121     void eject() {
122         this->~graduate();
123     }
124 private:
125
126 };
127 class TA :public teacher,public graduate {
128 public:
129     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) {
130         cout << "TA constructed" << endl;
131     }
132     TA(const TA& r):People(r),teacher(r),graduate(r) {
133
134     }
135     ~TA() {
136         cout << "TA destrcted" << endl;
137     }
138     inline void show() {
139         graduate::show();
140         cout << "principalship: " << principalship << " " <<
"department: " << department << " ";
141     }
142     void eject() {
143         this->~TA();
144     }
145 private:
146
147 };
148
149 int main() {
150     TA ta("alice", "1", "wo", "12345678", 2003, 1,
1, "123456", "electronic", "yang fang", "ta", "department of ee");
151     ta.show();
152
153
154 }
155

```

```

156 ostream& operator<<(ostream& o, People::Date& d) {
157     o << d.year << "年 " << d.month << "月 " << d.day << "日 ";
158     return o;
159 }
160

```

```

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 department: department of ee TA destroyed
graduate destroyed
student destroyed
teacher destroyed
people destroyed

```

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

```

1 #include <iostream>
2 using namespace std;
3 class People {
4 public:
5     class Date {
6     public:
7         Date(int y, int m, int d) :year(y), month(m), day(d) {}
8         ~Date() {}
9         int year, month, day;
10        friend ostream& operator <<(ostream& o, Date& d);
11    private:
12    };
13    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) {
14        strcpy_s(this->name, 11, name);

```

```

15         strcpy_s(this->number, 7, number);
16         strcpy_s(this->sex, 3, sex);
17         strcpy_s(this->id, 16, id);
18         cout << "People constructed" << endl;
19     }
20     ~People() {cout << "people destrcted" << endl;}
21     People(const People& r) :birthday(r.birthday) {
22         strcpy_s(this->name, 11, r.name);
23         strcpy_s(this->number, 7, r.number);
24         strcpy_s(this->sex, 3, r.sex);
25         strcpy_s(this->id, 16, r.id);
26     }
27     inline void show() {
28         cout << "name: " << name << " " << "number: " << number
<< " " << "sex: " << sex << " " << "id: " << id << " " <<
"birthday: " << birthday;
29     }
30     char name[11];
31     char number[7];
32     char sex[3];
33     char id[16];
34     Date birthday;
35     void eject() {this->~People();}
36 private:
37 };
38 class teacher :public People {
39 public:
40     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) {
41         strcpy_s(this->principalship, 11, ps);
42         strcpy_s(this->department, 21, de);
43         cout << "teacher constructed" << endl;
44     }
45     teacher(const teacher& r) :People(r) {
46         strcpy_s(this->principalship, 11, r.principalship);
47         strcpy_s(this->department, 21, r.department);
48     }
49     ~teacher() {cout << "teacher destrcted" << endl;}
50     char principalship[11];
51     char department[21];
52     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 destrcted" << 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 destrcted" << 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 destrcted" << 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