

程设第十五次作业

1. 分析访问属性

1. 在main函数中，可以使用b1.i访问基类成员 b1.j,b1.k不可因为公有继承方式，基类公有在派生中访问属性仍是公有，可以在类外访问，保护在派生类中仍是保护，不可在类外访问，而私有无法访问。
2. B的成员函数f3可以调用A中的f1() f2()
3. A中f1 f2函数在派生类中分别是公有和保护，都可以在派生类内访问
4. B中成员函数可引用A中的i j 不可引用k
5. A中i j在派生类中为公有和保护，可访问 而k在A中私有，不可在派生类中访问
6. main函数可以访问c1.i 其他均不可
7. main函数可以访问c1.f1() c1.f3() c1.f4() 不可访问c1.f2()
8. f4()中调用f1() f2() f3()均可

2. 分析程序

```
1  #include<iostream>
2  using namespace std;
3  class A{
4      public:
5          A() { //构造函数重载
6              a=0; b=0;
7          }
8          A(int i){
9              a=i; b=0;
10         }
11         A(int i,int j){
12             a=i; b=j;
13         }
14         void display(){
15             cout<<"a="<<a<<" b="<<b;
16         }
17     private:
18         int a;
19         int b;
20 };
21 class B:public A{
22     public:
23         B(){
24             c=0;
25         }
26         B(int i):A(i) { //派生类构造函数
27             c=0;
28         }
29         B(int i,int j):A(i,j){
30             c=0;
31         }
32         B(int i,int j,int k):A(i,j){
33             c=k;
34         }
35         void display1(){
```

```

36         display();
37         cout<<" c="<<c<<endl;
38     }
39     private:
40         int c;
41 };
42 int main(){
43     B b1;
44     B b2(1);
45     B b3(1,3);
46     B b4(1,3,5);
47     b1.display1();
48     b2.display1();
49     b3.display1();
50     b4.display1();
51     return 0;
52 }

```

运行结果如下:

```

a=0 b=0 c=0
a=1 b=0 c=0
a=1 b=3 c=0
a=1 b=3 c=5

```

3. 分析程序

```

1  #include<iostream>
2  using namespace std;
3  class A{
4      public:
5          A(){
6              cout<<"constructing A"<<endl;
7          }
8          ~A(){
9              cout<<"destructing A"<<endl;
10         }
11 };
12 class B:public A{
13     public:
14         B(){
15             cout<<"constructing B"<<endl;
16         }
17         ~B(){
18             cout<<"destructing B"<<endl;
19         }
20 };
21 class C:public B{
22     public:
23         C(){
24             cout<<"constructing C"<<endl;
25         }
26         ~C(){
27             cout<<"destructing C"<<endl;
28         }

```

```
29     };  
30     int main(){  
31         C c1;  
32         return 0;  
33     }  
34     //P356 调用派生类构造函数时会自动首先调用基类的构造函数，  
35     //而对派生类新增成员清理时，正好相反，先调用子对象的析构函数再调用基类的析构函数
```

运行结果如下：

```
constructing A  
constructing B  
constructing C  
destructing C  
destructing B  
destructing A
```