实验报告

P374T5

（1）

B1.i和b1.j正确，因为i和j分别是基类A中公有成员和受保护成员继承到派生类B中依然是公有和受保护状态。

b1.k错误，因为k是私有成员，虽然被继承到派生类中但是不能访问。

（2）

可以调用，派生类b继承的是基类a中的所有函数，只要是非私有的就可以访问。

（3）

i和j可以，k为基类私有，不可访问。

（4）

可以引用基类A的成员i，j，派生类B的成员m，派生类C的成员p，不能访问基类A中的私有成员k，派生类B的私有成员n。

（5）

可以调用，c为public继承，可以访问受保护和公有成员。

（6）

只要是非私有都可以访问，所以能调用。

P374T7

实验目的：

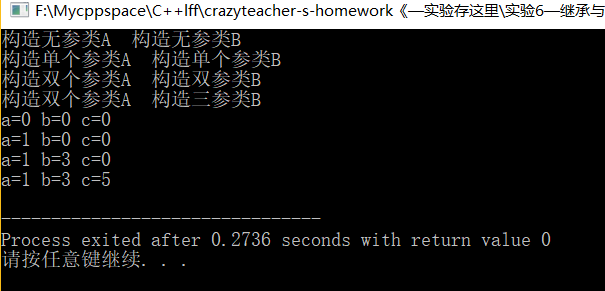
实现基础的派生类

了解派生类中对基类的构造的调用。

实验代码：

|  |
| --- |
| #include<iostream>  using namespace std;  class A  {  public:  A(){a=0;b=0;cout<<"构造无参类A ";}  A(int i){a=i;b=0;cout<<"构造单个参类A ";}  A(int i,int j){a=i;b=j;cout<<"构造双个参类A ";}  void display(){cout<<"a="<<a<<" b="<<b;}  private:  int a;  int b;  };  class B : public A  {  public:  B(){c=0;cout<<"构造无参类B\n";}  B(int i):A(i){c=0;cout<<"构造单个参类B\n";}  B(int i,int j):A(i,j){c=0;cout<<"构造双参类B\n";}  B(int i,int j,int k):A(i,j){c=k;cout<<"构造三参类B\n";}  void display1()  {  display();  cout<<" c="<<c<<endl;  }  private:  int c;  };  int main()  {  B b1;  B b2(1);  B b3(1,3);  B b4(1,3,5);  b1.display1();  b2.display1();  b3.display1();  b4.display1();  return 0;  } |

运行结果:



结果分析：

由结果可以看出main函数中定义的b1,b2,b3,b4四个对象分别调用了基类A中的无参，单个参数，两个参数，两个参数的构造函数，定义对象时先调用基类构造函数，再调用派生类构造函数。

P374T9

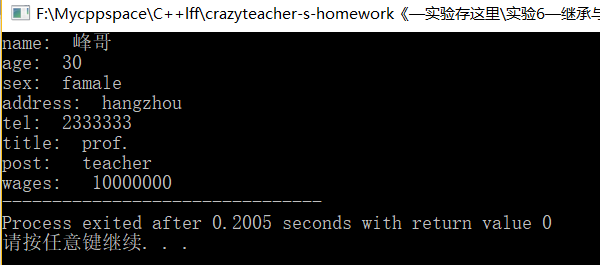
实验目的：

学会使用多重继承。

实验代码

|  |
| --- |
| #include<iostream>  #include<string>  using namespace std;  class teacher  {  public:  teacher(string nam,int a,string se,  string addr,string t,string tit);  void display();  protected:  int age;  string name;  string sex;  string address;  string tel;  string title;  };  teacher::teacher(string nam,int a,string se,  string addr,string t,string tit):  name(nam),age(a),sex(se),address(addr),  tel(t),title(tit){}  void teacher::display()  {  cout<<"name"<<name<<endl;  cout<<"age"<<age<<endl;  cout<<"sex"<<sex<<endl;  cout<<"address"<<address<<endl;  cout<<"tel"<<tel<<endl;  cout<<"title"<<title<<endl;  }  class cadre  {  public:  cadre(int a,string nam,string pos,string se,string addr,string t);  void display();  protected:  int age;  string post;  string name;  string sex;  string address;  string tel;  };  cadre::cadre(int a,string nam,string pos,string se,string addr,string t):  age(a),name(nam),post(pos),sex(se),address(addr),tel(t){}  void cadre::display()  {  cout<<"name"<<name<<endl;  cout<<"age"<<age<<endl;  cout<<"post"<<post<<endl;  cout<<"sex"<<sex<<endl;  cout<<"address"<<address<<endl;  cout<<"tel"<<tel<<endl;  }  class teacher\_cadre:public teacher,public cadre  {  public:  teacher\_cadre(int age,string name,string post,string sex,  string address,string tel,string title,int wages);  void show();  private:  int wages;  };  teacher\_cadre::teacher\_cadre(int age,string name,string post,  string sex,string address,string tel,string title,int wage):  teacher(name,age,sex,address,tel,title),  cadre(age,name,post,sex,address,tel),wages(wage){}  void teacher\_cadre::show()  {  teacher::display();  cout<<"post"<<cadre::post<<endl;  cout<<"wages"<<wages;  }  int main()  {  teacher\_cadre c(30,"峰哥","teacher","famale","hangzhou","2333333","prof.",10000000);  c.show();  return 0;  } |

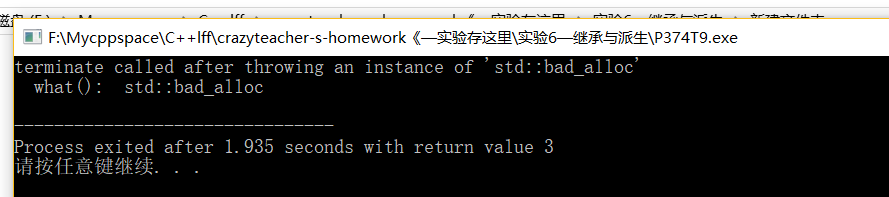
运行结果：

结果分析：

对两个多个基类的继承要注意不要搞混同名函数

按序给数据成员赋值输出。

出现过的错误



在写派生类的时候少了一个数据成员，初始化后输出的时候内存爆了。

写代码一定要细致！

P374T10

实验目的：

实现类的继承与组合

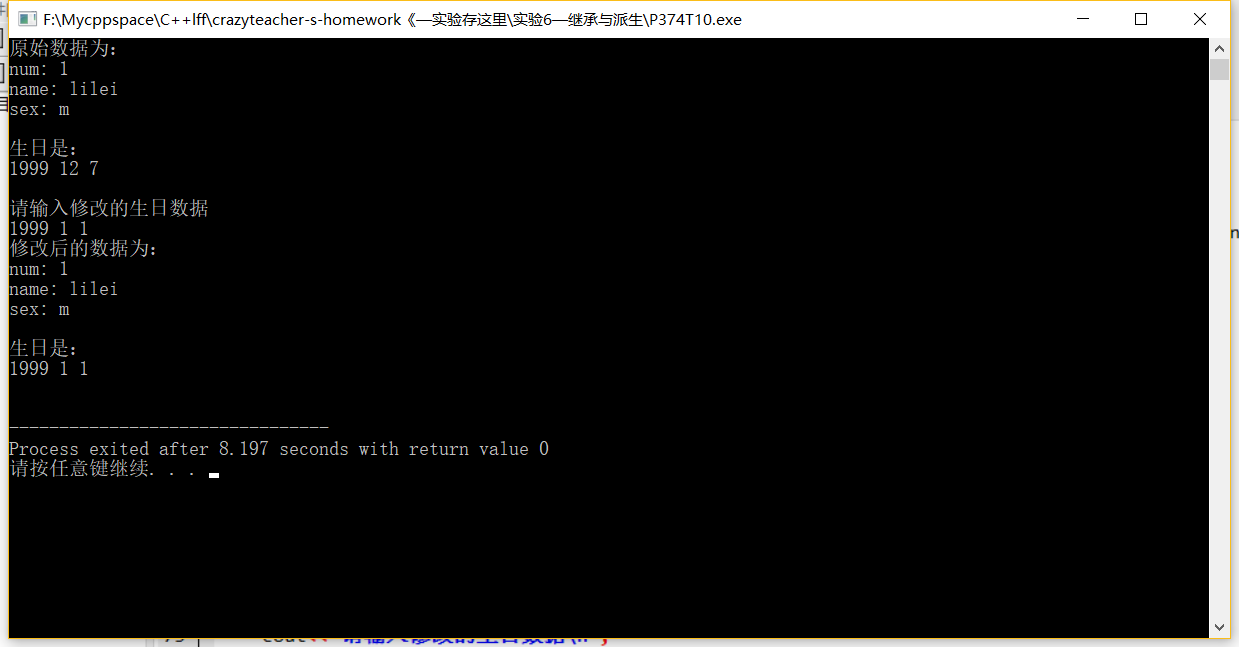
同名函数的使用

数据可修改

实验代码：

|  |
| --- |
| #include<iostream>  #include<cstring>  #include<string>  using namespace std;  class teacher  {  public:  teacher(int num,string name,char sex);  void display();  private:  int num;  string name;  char sex;  };  teacher::teacher(int num,string name,char sex):  num(num),name(name),sex(sex){}  void teacher::display()  {  cout<<"num: "<<num<<endl  <<"name: "<<name<<endl  <<"sex: "<<sex<<endl;  }  class birthdate  {  public:  birthdate(int year,int month,int day);  void display();  void reset(int y,int m,int d);  private:  int year;  int month;  int day;  };  birthdate::birthdate(int year,int month,int day):  year(year),month(month),day(day){}  void birthdate::display()  {  cout<<"生日是：\n";  cout<<year<<' '<<month<<' '<<day;  }  void birthdate::reset(int y,int m,int d)  {  year = y;  month = m;  day = d;  }  class professor:public teacher  {  public:  professor(int num,string name,char sex,int year,int month,int day);  void display();  void resetbirthdate(int y,int m,int d);  private:  birthdate birthday;  };  professor::professor(int num,string name,char sex,int year,int month,int day):teacher(num,name,sex),birthday(year,month,day){}  void professor::display()  {  teacher::display();  cout<<endl;  birthday.display();  cout<<endl<<endl;  }  void professor::resetbirthdate(int y,int m,int d)  {  birthday.reset(y,m,d);  }  int main()  {  cout<<"原始数据为：\n";  professor prof1(1,"lilei",'m',1999,12,7);  prof1.display();  cout<<"请输入修改的生日数据\n";  int y,m,d;  cin>>y>>m>>d;  cout<<"修改后的数据为：\n";  prof1.resetbirthdate(y,m,d);  prof1.display();  } |

运行结果



结果分析：

实现了对生日数据的修改，并同步到profession类中，并输出。