- animal.h

```
#ifndef ANIMAL_H
#define ANIMAL_H
class Animal{
public:
    /*如果没有这个public的构造函数来初始化age,不能通过dog的对象对age进行初始化
        不论哪种继承方式，private的成员都不能被派生类继承
    */
    Animal(int age):age(age){}
    int getAge() const{return age;}
private:
    int age;
};
#endif
```

- dog.h

```
#ifndef DOG_H
#define DOG_H
#include"Animal.h"
class dog:public Animal{
public:
    dog(int age):Animal(age){}

};
#endif
```

- BaseClass.h

```
#ifndef BASECLASS_H
#define BASECLASS_H
#include <iostream>
using namespace std;
class BaseClass{
public:
    BaseClass(int number):number(number){cout <<
"BaseCLass构造函数被调用"<< endl;}
    ~BaseClass(){cout <<"BaseClass析构函数被调用"<<
endl;}
private:
    int number;
};

#endif
```

- DerivedClass.h

```
#ifndef DERIVEDCLASS_H
#define DERIVEDCLASS_H
#include"BaseClass.h"
class DerivedClass:public BaseClass{
public:
    DerivedClass(int number):BaseClass(number)
{std::cout << "DerivedClass 构造函数被调用"<<
std::endl;}
    ~DerivedClass(){std::cout << "DerivedClass析构函
数被调用" <<std::endl;}
};

#endif
```

- vehicle.h

```
#ifndef VEHICLE_H
#define VEHICLE_H
class vehicle{
public:
    void Run(){};
    void Stop(){};
private:
    double MaxSpeed;
    int Weight;
};
#endif
```

- bicycle.h

```
#include"vehicle.h"
#ifndef BICYCLE_H
#define BICYCLE_H
class bicycle:virtual public vehicle{
private:
    int Height;
};
#endif
```

- motorcar.h

```
#include"vehicle.h"
#ifndef MOTORCAR_H
#define MOTORCAR_H
//如果不用虚继承，run和stop函数在不同类的对象中会出现二义性，编译报错
class motorcar:virtual public vehicle{
public:
    int getSeatNum() const{return SeatNum;}
private:
    int SeatNum;
};
#endif
```

- motobicycle.h

```
#include "motorcar.h"
#include"bicycle.h"
#ifndef MOTORCYCLE_H
#define MOTORCYCLE_H
class motorcycle:public motorcar,public bicycle{

};
#endif
```

- main.cpp

```
#include "dog.h"
#include"DerivedClass.h"
#include <iostream>
#include <cstdlib>
#include"bicycle.h"
#include"motorcar.h"
#include"motorcycle.h"
using namespace std;
int main(){
    cout << "----------这是第一题--------" << endl;
    int age;
    cin >> age;
    dog d(age);
    cout << d.getAge()<< endl;
    cout << "----------这是第二题--------" << endl;
    int number;
    cin >> number;
    DerivedClass derive(number);
    derive.~DerivedClass();
    cout << "----------这是第三题--------" << endl;
    bicycle bi;
    bi.Run();
    motorcar mocar;
    mocar.Run();
    motorcycle mocycle;
```

```
    mocycle.Run();
    system("pause");
    return 0;
}
```

- 运行结果

```
----------这是第一题--------
3
3
----------这是第二题--------
200
BaseCLass构造函数被调用
DerivedClass 构造函数被调用
DerivedClass析构函数被调用
BaseClass析构函数被调用
----------这是第三题--------
请按任意键继续. . .

DerivedClass析构函数被调用
BaseClass析构函数被调用
```