实验 7：继承

姓名：陈王子

班级：大数据分析 2101 班

学号：202103150503

- **请阅读此说明：实验 7 满分 100 分。做完实验后请按要求将代码和截图贴入该文档。然后将此文档、源代码文件（.hpp，.cpp）打包上传到学习通。**

实验目的：熟悉并掌握继承机制，能够利用公有继承方式建立符合用户需求的类族。

实验要求：按照每个类两个文件的方式（一个头文件，一个源文件）组织工程内的代码。

实验内容：

1、请仔细观察下列类声明，并回答：

```
 class A { //基类
 public:
      A(int v1=0,int v2=0,int v3=0):a(v1),b(v2),c(v3){ }
      void F1(){cout<< "F1"<<a<<" "<<b<<" "<<c<<endl;}
     int a;
protected:
      void F2( ) {cout<<"F2"<<a<<" "<<b<<" "<<c<<endl;}
     int b;
```

```
private:

      void F3(){cout<<"F3"<<a<<" "<<b<<" "<<c<<endl;}

     int c;

};

class B: public A{

public:

    //B 的构造函数缺失

    void F4( ) {cout<<"F4"<<Ba<" "<<Bb<<" "<<Bc<<endl;}

    int Ba;

protected:

    void F5( ) {cout<<"F5"<<Ba<<" "<<Bb<<" "<<Bc<<endl;}

    int Bb;

private:

    void F6(){cout<<"F6"<<Ba<<" "<<Bb<<" "<<Bc<<endl;}

    int Bc;

};

class C: protected  B{

public:

   //C 的构造函数缺失

   void F7(){cout<<"F7"<<Ba<<" "<<Bb <<endl;}

   void F8(){cout<<"F8"<<Ca<<" "<<Cb <<endl;}

   int Ca;
```

```
private:
   int Cb;
};


//测试主函数
int main()
{
    A Aobj1,Aobj2(1,2,3);
    B Bobj1,Bobj2(1,2,3,4,5);
    C Cobj1,Cobj2(1,2,3,4,5,6);
    ......
    return 0;
}
```

(1) 填写表格，写出第一行标识符在第一列所展示的各个作用域的访问控制方式(public,protected,private)。(10 分)

访问域\成员名	a	b	c	F1	F2	F3	Ba	Bb
A	public	protected	private	public	protected	private	-	-
B	public	protected	private	public	protected	private	public	protected

C	protected	protected	private	protected	protected	private	public	protected
main 函数	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
F4	F5	F6	Ca	Cb	F7			
-	-	-	-	-				
public	protected	private	-	-				
public	protected	private	protected	private	public			
N/A	N/A	N/A	N/A	N/A	N/A			

(2) 补充完类 B 和类 C 缺失的构造函数，并将 main 的测试程序补充完整。要求在 main 中展示类 A，类 B，类 C 的所有可在 main 中访问的成员。(40 分)

- **补充 B 的构造函数:**

```
class B: public A{
public:
    B(int v1=0, int v2=0, int v3=0, int v4=0, int v5=0) : A(v1, v2, v3), Ba(v4), Bb(v5), Bc(0) {}
    void F4( ) {cout<<"F4"<<Ba<<" "<<Bb<<" "<<Bc<<endl;}
    int Ba;
protected:
    void F5( ) {cout<<"F5"<<Ba<<" "<<Bb<<" "<<Bc<<endl;}
    int Bb;
private:
    void F6(){cout<<"F6"<<Ba<<" "<<Bb<<" "<<Bc<<endl;}
    int Bc;
};
```

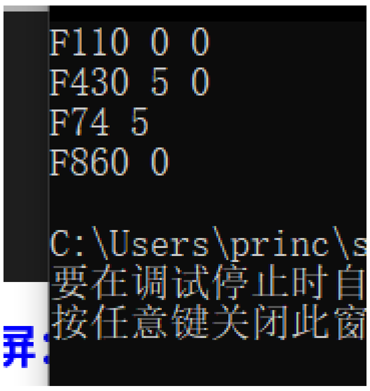
- **补充 C 的构造函数:**

```
class C : protected B {
public:
    C(int v1 = 0, int v2 = 0, int v3 = 0, int v4 = 0, int v5 = 0, int v6 = 0) : B(v1, v2, v3, v4, v5), Ca(v6), Cb(0) {}
    void F7() { cout << "F7" << Ba << " " << Bb << endl; }
    void F8() { cout << "F8" << Ca << " " << Cb << endl; }
    int Ca;
protected:
    int Cb;
};
```

- **main 函数：**

```

int main()
{
    A Aobj1, Aobj2(1, 2, 3);
    B Bobj1, Bobj2(1, 2, 3, 4, 5);
    C Cobj1, Cobj2(1, 2, 3, 4, 5, 6);

    // 访问类 A 的成员
    Aobj1.a = 10;
    Aobj1.F1();

    // 访问类 B 的成员
    Bobj2.a = 20;
    Bobj2.Ba = 30;
    Bobj2.F4();

    // 访问类 C 的成员
    Cobj2.Ca = 60;
    Cobj2.F7();
    Cobj2.F8();

    return 0;
}
```

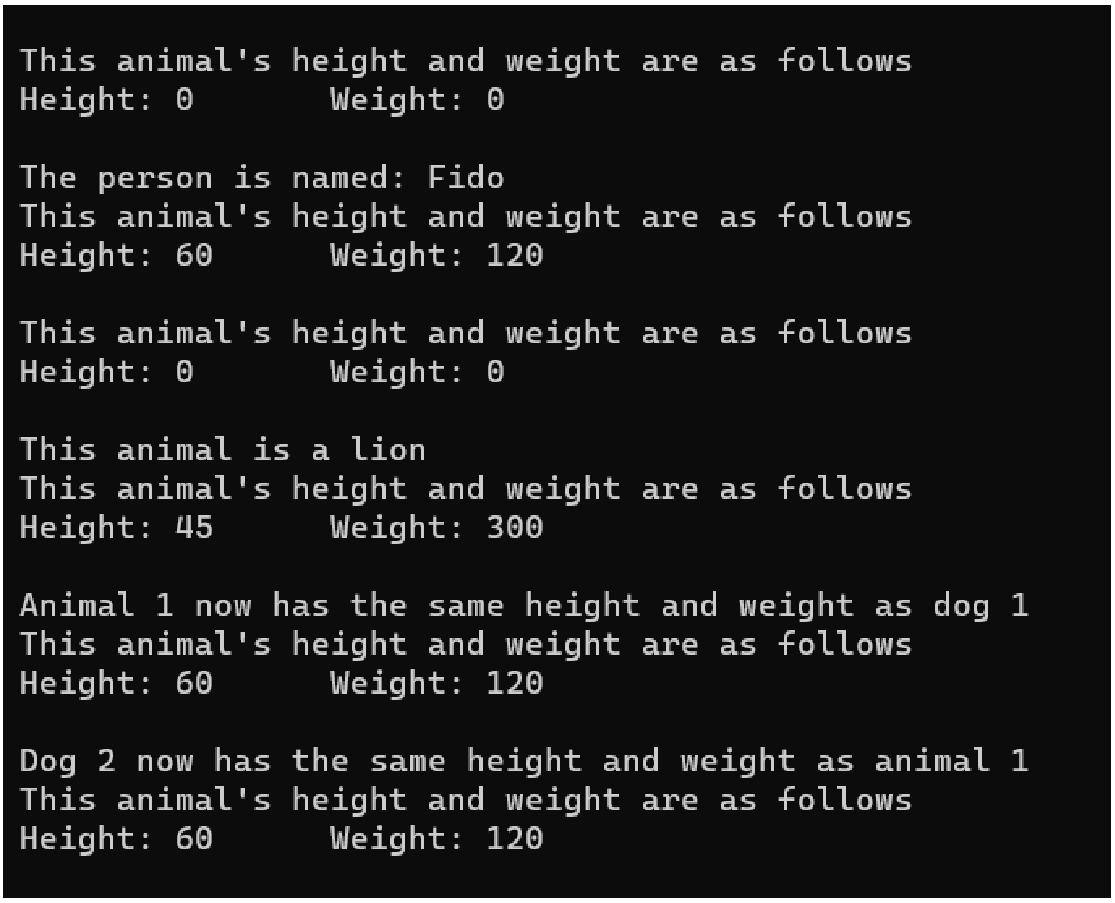
- **程序运行结果截屏：**



```
F110 0 0
F430 5 0
F74 5
F860 0

C:\Users\princ\s
要在调试停止时自
按任意键关闭此窗
```

2、代码调试：附件中的代码在建立类族的过程中，由于编程人员的疏忽，出现了一些小问题，请帮忙修改过来。（20 分）

- **修改后的代码运行结果截屏：**



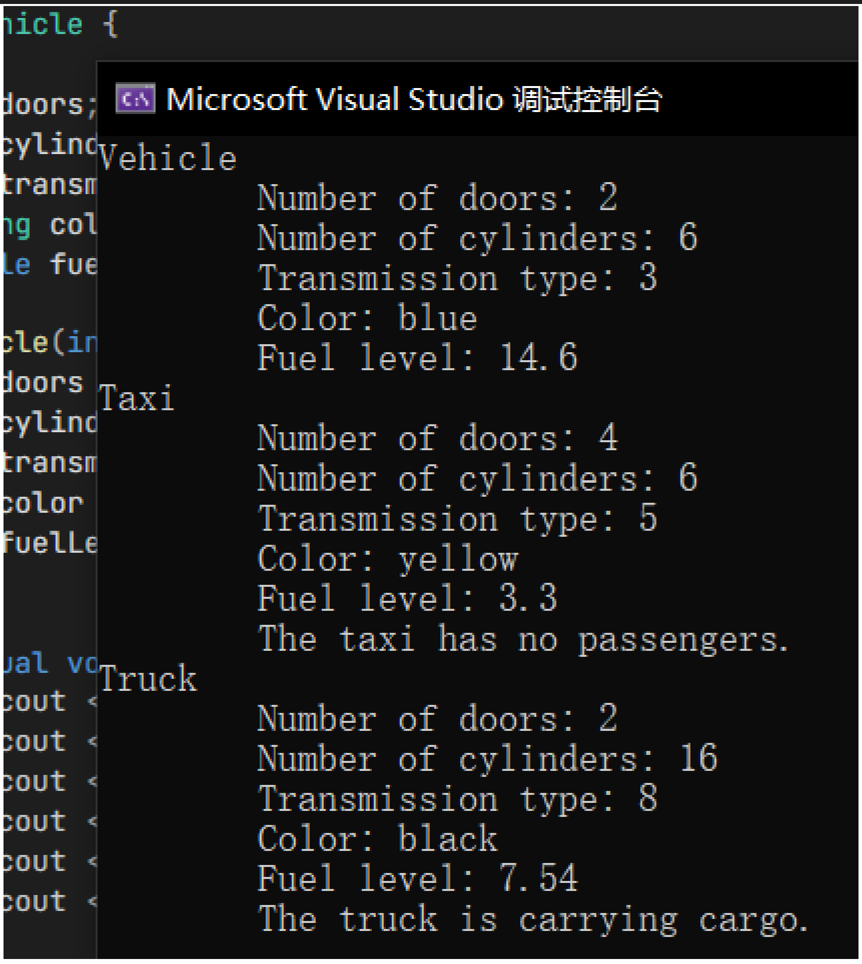
```
This animal's height and weight are as follows
Height: 0        Weight: 0

The person is named: Fido
This animal's height and weight are as follows
Height: 60       Weight: 120

This animal's height and weight are as follows
Height: 0        Weight: 0

This animal is a lion
This animal's height and weight are as follows
Height: 45       Weight: 300

Animal 1 now has the same height and weight as dog 1
This animal's height and weight are as follows
Height: 60       Weight: 120

Dog 2 now has the same height and weight as animal 1
This animal's height and weight are as follows
Height: 60       Weight: 120
```

3、设计交通工具类族： 开发一个名为 Vehicle 的类的层次体系。创建两个类 Taxi 和 Truck，均以公有模式从类 Vehicle 中继承而来。Taxi 类中应包含一个数据成员 passenger 说明其是否载客。Truck 类应包含一个数据成员 cargo 说明其是否载货。根据题后附的测试程序输出结果 为类 Vehicle 添加必要的数据成员,并为所有类添加必要的函数来控制和访问类的数据。编写一段测试程序，将 Vehicle 对象、Truck 对象和 Taxi 对象打印到屏幕。(30 分)

```
#include <iostream>
#include <string>
using namespace std;

class Vehicle {
public:
```

```cpp
    int doors;
    int cylinders;
    int transmissionType;
    string color;
    double fuelLevel;

    Vehicle(int _doors, int _cylinders, int _transmissionType, string
_color, double _fuelLevel) {
        doors = _doors;
        cylinders = _cylinders;
        transmissionType = _transmissionType;
        color = _color;
        fuelLevel = _fuelLevel;
    }

    virtual void print() {
        cout << "Vehicle" << endl;
        cout << "\tNumber of doors: " << doors << endl;
        cout << "\tNumber of cylinders: " << cylinders << endl;
        cout << "\tTransmission type: " << transmissionType << endl;
        cout << "\tColor: " << color << endl;
        cout << "\tFuel level: " << fuelLevel << endl;
    }
};

class Taxi : public Vehicle {
public:
    bool passenger;

    Taxi(int _doors, int _cylinders, int _transmissionType, string
_color, double _fuelLevel, bool _passenger) : Vehicle(_doors,
_cylinders, _transmissionType, _color, _fuelLevel) {
        passenger = _passenger;
    }

    void print() override {
        cout << "Taxi" << endl;
        cout << "\tNumber of doors: " << doors << endl;
        cout << "\tNumber of cylinders: " << cylinders << endl;
        cout << "\tTransmission type: " << transmissionType << endl;
        cout << "\tColor: " << color << endl;
        cout << "\tFuel level: " << fuelLevel << endl;
        if (passenger) {
            cout << "\tThe taxi has passengers." << endl;
        }
        else {
            cout << "\tThe taxi has no passengers." << endl;
        }
    }
};

class Truck : public Vehicle {
public:
    bool cargo;

    Truck(int _doors, int _cylinders, int _transmissionType, string
_color, double _fuelLevel, bool _cargo) : Vehicle(_doors, _cylinders,
_transmissionType, _color, _fuelLevel) {
        cargo = _cargo;
    }

    void print() override {
        cout << "Truck" << endl;
        cout << "\tNumber of doors: " << doors << endl;
        cout << "\tNumber of cylinders: " << cylinders << endl;
        cout << "\tTransmission type: " << transmissionType << endl;
        cout << "\tColor: " << color << endl;
        cout << "\tFuel level: " << fuelLevel << endl;
        if (cargo) {
            cout << "\tThe truck is carrying cargo." << endl;
        }
        else {
            cout << "\tThe truck is not carrying cargo." << endl;
        }
    }
};

int main() {
    Vehicle v(2, 6, 3, "blue", 14.6);
    v.print();

    Taxi t(4, 6, 5, "yellow", 3.3, false);
    t.print();

    Truck tr(2, 16, 8, "black", 7.54, true);
    tr.print();

    return 0;
```

```
}
```



```
nicle {

doors;
cylind
transm
ng col
le fue

cle(in
doors
cylind
transm
color
fuelLe


ual vo
cout <
cout <
cout <
cout <
cout <
cout <
```

```
Vehicle
        Number of doors: 2
        Number of cylinders: 6
        Transmission type: 3
        Color: blue
        Fuel level: 14.6
Taxi
        Number of doors: 4
        Number of cylinders: 6
        Transmission type: 5
        Color: yellow
        Fuel level: 3.3
        The taxi has no passengers.
Truck
        Number of doors: 2
        Number of cylinders: 16
        Transmission type: 8
        Color: black
        Fuel level: 7.54
        The truck is carrying cargo.
```