第7讲

多重继承

主讲人：赵文彬

本次课主要内容

- 声明多重继承的方式
- 多重继承派生类的构造函数
- 二义性问题
- 虚基类

声明多重继承的方式

```
class 派生类名：继承方式1 基类名1, 继承方式2 基类名2,…
{
   private:
          派生类的私有数据和函数
   public:
          派生类的公有数据和函数
   protected:
          派生类的保护数据和函数
};
```

多重继承派生类的构造函数

派生类构造函数的一般形式：

派生类构造函数名(总参数表)： 基类构造函数名(参数表),子对象名(参数表)

{

派生类中新增数据成员初始化

}

多继承中的二义性

- **由于多继承中派生类拥有多个基类，**如果多个基类中拥有同名的成员。

客货两用车类

```
#include<iostream>
using  namespace std;
class Car      //小客车类
{
private:
   int power;   //马力
   int seat;     //座位
public:
   Car(int power,int seat)
   {
      this->power=power;
      this ->seat=seat;
   }
   void show()
   {
     cout<<"car power:"<<power;
     cout<<" seat:"<<seat<<endl;
   }
};
```

客货两用车类

```
class Wagon //小货车类
{
private:
   int power;   //马力
    int load;     //装载量
public:
    Wagon(int power,int
load)
    {
        this->power=power;
        this->load=load;
    }
```

```
 void show()
   {
       cout<<"wagon
power:“<<power;
       cout<<"
load:"<<load<<endl;
   }
};
```

```
class StationWagon :public Car, public Wagon  //客货两
用车类
{
public:
   StationWagon(int power, int seat,int
load):Wagon(power,load),
   Car(power,seat){}

  void ShowSW()
  {
     cout<<"StationWagon:"<<endl;
     Car::show();
     Wagon::show();
  }
};
```

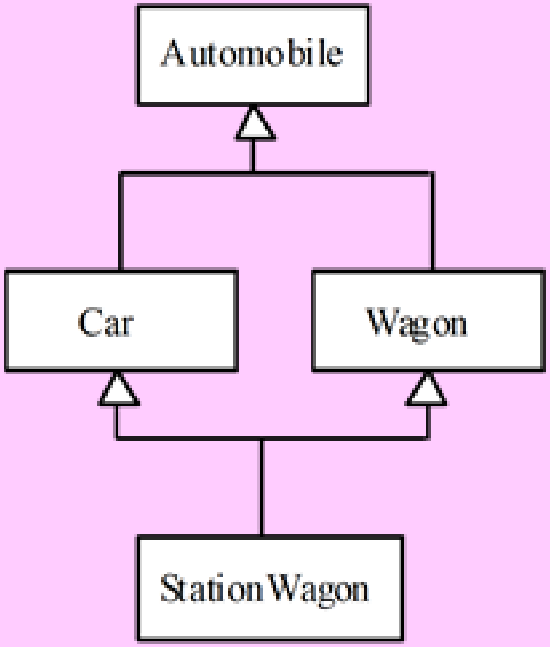
客货两用车类

```
void main()
{
    StationWagon SW(105,3,8);
    SW.show();        //错误，出现二义性
    SW.ShowSW();
}
```

二义性解决方法

- 加上域作用符::；例如，Car::show()和Wagon::show()。
- 在子类中定义函数覆盖父类中具有歧义性的函数，例如定义的函数showSW()。

多层次多继承中的歧义性



(a) 继承层次图

<table>
<tr><td rowspan="9">StationWagon::</td><td rowspan="4">Car::</td><td rowspan="2">Automobile::</td><td>power</td></tr>
<tr><td>show()</td></tr>
<tr><td colspan="2">seat</td></tr>
<tr><td colspan="2">show()</td></tr>
<tr><td rowspan="4">Wagon::</td><td rowspan="2">Automobile::</td><td>power</td></tr>
<tr><td>show()</td></tr>
<tr><td colspan="2">load</td></tr>
<tr><td colspan="2">show()</td></tr>
<tr><td colspan="3">show()</td></tr>
</table>

(b)成员构成图

客货两用车类2

```
#include<iostream>
using  namespace std;
class Automobile
//汽车类
{
private:
    int power;   //马力
public:
    Automobile(int power)
    {
        this->power=power;
    }
```

```
void show()
   {
        cout<<"  power:"<<power;
   }
};
```

客货两用车类2

```
class Car: public Automobile      //小
客车类
{
private:
     int seat;     //座位
public:
    Car(int power,int
seat):Automobile(power)
    {
        this->seat=seat;
    }
```

```
 void show()
   {
         cout<<"car:";
         Automobile::show();
         cout<<"
seat:"<<seat<<endl;
   }
};
```

客货两用车类2

```
class Wagon: public Automobile //
小货车类
{
private:
    int load;    //装载量
public:
    Wagon(int power,int
load):Automobile(power)
    {
        this->load=load;
    }
```

```
 void show()
    {
        cout<<"wagon:";
        Automobile::show();
        cout<<"
load:"<<load<<endl;
    }
};
```

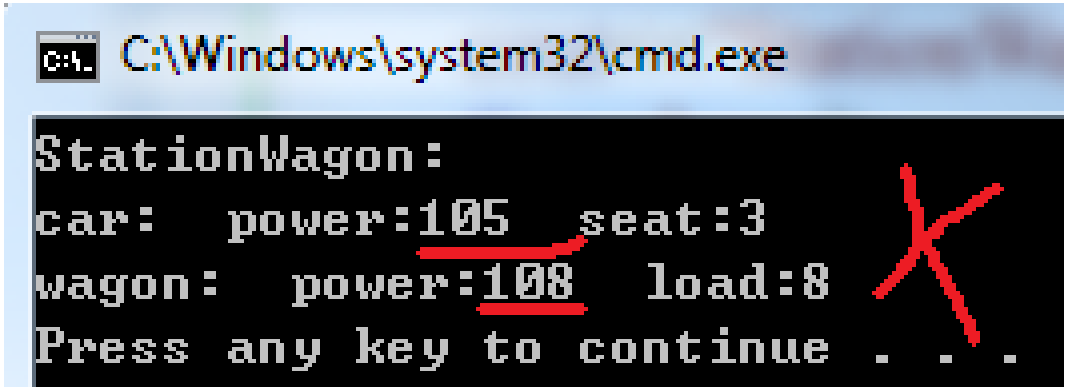
客货两用车类2

```
class StationWagon :public Car, public Wagon  //客货两用车类
{
public:
  StationWagon(int CPower, int WPower,int seat,int load):  Wagon(WPower, load), Car(CPower, seat)     {}
   void show()
   {
        cout<<"StationWagon:"<<endl;
        Car::show();
       Wagon::show();
    }
};
```

客货两用车类2

```
void main()
{
   StationWagon SW(105,108,3,8);
    SW.show();
}
```

二义性



```
StationWagon:
car:   power:105   seat:3
wagon:   power:108   load:8
Press any key to continue . . .
```

虚基类

class 派生类名： virtual 继承方式 基类名

客货两用车类3

```
#include<iostream>
using  namespace std;
class Automobile
{
private:
    int power;  //马力
public:
    Automobile(int power)
    {
        this->power=power;
        cout<<"Automobile constructing..."<<endl;
    }
```

```
void show()
   {
        cout<<" power:"<<power;
   }
};
```

客货两

```
class Car: virtual public A
类
{
private:
   int seat;     //座位
public:
  Car(int power,int seat):Automobile(power)
  {
     this->seat=seat;
    cout<<"Car constructing..."<<endl;
  }
```

```
void show()
{
    cout<<"car:";
    Automobile::show();
    cout<<"  seat:"<<seat<<endl;
 }
};
```

客

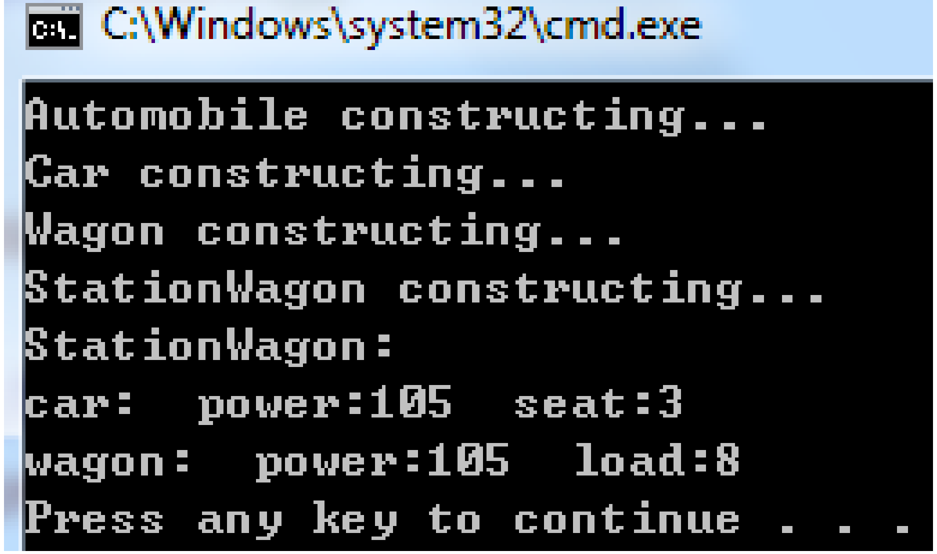
```
class Wagon: virtual p
{
private:
     int load;     //装载量
public:
    Wagon(int power,int load):Automobile(power)
    {
       this->load=load;
       cout<<"Wagon constructing..."<<endl;
    }
```

```
void show()
    {
          cout<<"wagon:";
         Automobile::show();
          cout<<"  load:"<<load<<endl;
    }
};
```

```
class StationWagon :public Car, public Wagon  //客货两用车类
{
public:
  StationWagon(int CPower,int WPower, int seat,int
load)    :Automobile(CPower),Wagon(WPower,load),
Car(CPower,seat)  {
    cout<<"StationWagon constructing..."<<endl;
 }
 void show()
 {
    cout<<"StationWagon:"<<endl;
    Car::show();
    Wagon::show();
 }
};
```

客货两用车类3

```
void main()
{
   StationWagon SW(105,108,3,8);
   SW.show();
}
```



```
Automobile constructing...
Car constructing...
Wagon constructing...
StationWagon constructing...
StationWagon:
car:  power:105  seat:3
wagon:  power:105  load:8
Press any key to continue . . .
```

课堂练习

学生具有姓名，班级，学号等属性，有上课等行为；教师具有姓名，工号，工资等属性，有教课等行为；助教既是学生，又是老师，具有学生和老师的双重属性。请用类的多继承机制实现上述问题。

定义People类，派生学生类和教师类，再由学生类和教师类派生助教类。

本次课小结

- 理解二义性产生原因
- 掌握虚基类使用方法