河北师范大学软件学院
Software College of Hebei Normal University

C++

第十三讲 继承与派生（三）

基础课教研室C++ 课程组

上一讲教学目标

- 掌握C++中单重继承中构造函数的调用
- 掌握C++中单重继承中析构函数的调用

本讲教学目标

- 了解基类和派生类间的赋值兼容规则
- 了解同名冲突及其解决方案

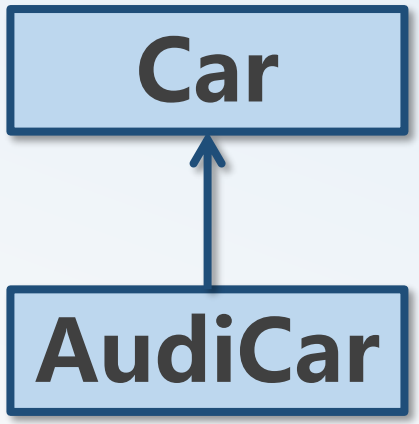
1 基类和派生类间的赋值兼容规则

2 同名冲突及其解决方案

同名隐藏

基类和派生类间的赋值兼容规则

- 在公有继承方式下，基类和派生类间存在赋值兼容，具体赋值规则
  - 基类对象 = 公有派生类对象(仅能访问基类部分)
  - 基类对象的指针 = 公有派生类对象的地址（同上）
  - 基类的引用 = 公有派生类对象（同上）




```
Car      carObj;
AudiCar audiObj;
carObj   = audiObj;
Car * pB    = &audiObj;
Car & refB = audiObj;
```

基类和派生类间的赋值兼容规则

```
class Car {
public:
    Car(){}
    void display() const {
      cout<<"Car" << endl;
    }
};
class AudiCar : public Car {
public:
    AudiCar():Car() {}
    void display() const {
       cout<<"AudiCar"<<endl;
    }
};
```

```
int main(void) {
//基类对象访问的一定是基类成员
    Car objCar;
    objCar.display();
//基类指针指向基类对象,只能访问基类成员
    Car * pCar = &objCar;
    pCar->display();
//基类指针指向派生类对象,只能访问基类成员
    AudiCar objDer;
    pCar = &objDer;
    pCar->display();
//基类引用作为派生类对象的别名,只能访问基类成员
    Car & obj = objDer;
    obj.display();
    return 0; }
```

基类和派生类间的赋值兼容规则

- 在公有继承方式下，赋值兼容规则的说明：
  - 基类对象只能访问基类的成员
  - 用基类的指针，无论是否指向基类对象，都只能访问基类部分的成员
  - 用基类的引用，无论是否是基类对象的别名，都只能访问基类部分的成员
  - 这是一种向上的类型转换，是安全的。

1 基类和派生类间的赋值兼容规则

2 同名冲突及其解决方案

同名隐藏

同名隐藏

- 在派生类里如果存在与基类同名的成员，访问派生类里的同名成员时该如何处理？
  - 数据成员同名、成员函数同名

```
class Car {
public:
    void test() {
        cout << "Car";
    }
protected:
    int m_iVal;
};
```

```
class AudiCar : public Car {
public:
    void test() {
        cout << "AudiCar";
    }
protected:
    int m_iVal;
};
```

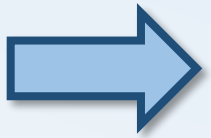
同名隐藏

❖ 同名隐藏

- 若派生类成员和基类成员重名，在派生类中使用的是派生类的同名成员，基类的同名成员自动被隐藏。
- 若要在派生类中访问基类中被隐藏同名成员（如果允许访问），可以使用基类名限定。

基类名::数据成员
基类名::成员函数名（实参）

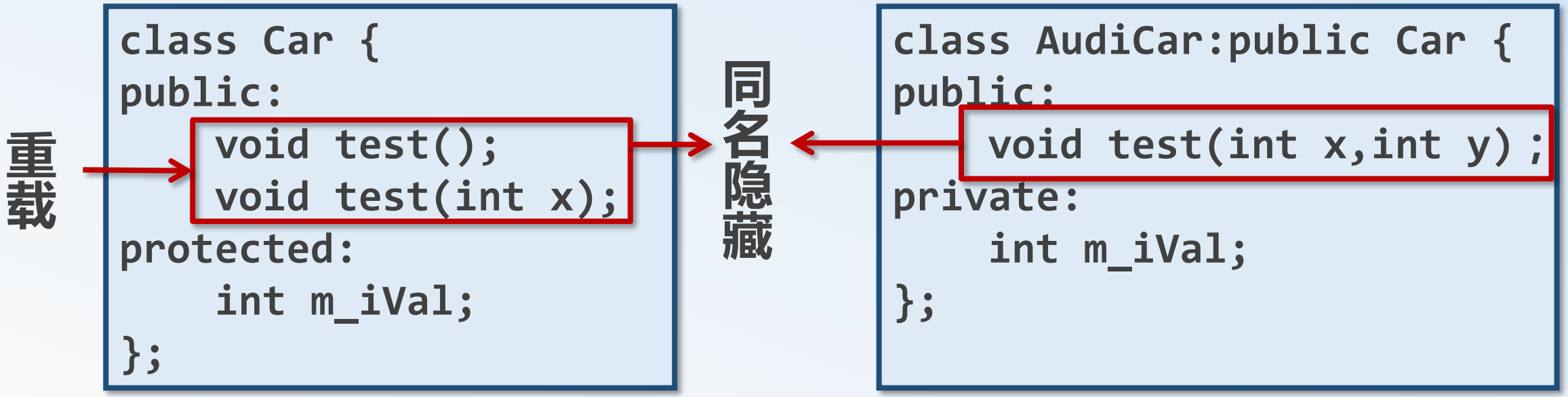
AudiCar



```
test();
m_ival;
Car::test();
Car::m_ival;
```

同名隐藏

- 如果基类和派生类中有同名函数，那么这个同名函数是函数重载吗？
  - 函数重载的条件是同一作用域，派生类和基类分别属于不同的作用域，所以派生类和基类中同名函数的不叫重载叫做同名隐藏。



本讲教学目标

- 了解基类和派生类间的赋值兼容规则
- 了解同名冲突及其解决方案


THANKS