【C++】Day77

Class	C++
Date	@March 26, 2022
Material	
Series Number	
Summary	

【Ch15】OOP

15.7.3 Derived-Class Copy-Control Members

The copy and move constructors for a derived class must copy/move the members of its base part as well as the members in the derived.

Unlike the constructors and assignment operators, the destructor is responsible only for destroying the resources allocated by the derived class.

Warning: When a derived class defines a copy or move operation, that operation is responsible for copying or moving the entire object, including base-class members.

Defining a Derived Copy or Move Constructor

When we define a copy or move constructor for a derived class, we ordinarily use the corresponding base-class constructor to initialize the base part of the object:

```
class Base { /* ... */};

class D : public Base {
  // By default, the base class default constructor initializes the base part of an object
  // to use the copy or move constructor, we must explicitly call that
  // constructor in the constructor initializer list
  D(const D& d) : Base(d) // copy the base members
    /* Initializers for members of D */ { /* ... */ }
  D(D&& d) : Base(std::move(d)) // move the base members
    /* Initializers for members of D */ { /* ... */ }
};
```

Warning: By default, the base-class default constructor initializes the base-class part of a derived object. If we want copy (or move) the base-class part, we must explicitly *use the copy (or move) constructor for the base class in the derived’s constructor initializer list.*