

第16章 override和final说明符

《现代C++语言核心特性解析》 谢丙堃

分清重写、重载和隐藏

- 重载（overload），它通常是指在同一个类中有两个或者两个以上函数，它们的函数名相同，但是函数签名不同，也就是说有不同的形参。
- 重写（override）的意思更接近覆盖，在C++中是指派生类覆盖了基类的虚函数
- 隐藏（overwrite）是指基类成员函数，无论它是否为虚函数，当派生类出现同名函数时，如果派生类函数签名不同于基类函数，则基类函数会被隐藏。如果派生类函数签名与基类函数相同，则需要确定基类函数是否为虚函数，如果是虚函数这里的概念就是重写，否则基类函数也会被隐藏。

重写引发的问题

- 例子

```
class Base {  
public:  
    virtual void some_func() {}  
    virtual void foo(int x) {}  
    virtual void bar() const {}  
    void baz() {}  
};  
class Derived : public Base {  
public:  
    virtual void sone_func() {}  
    virtual void foo(int &x) {}  
    virtual void bar() {}  
    virtual void baz() {}  
};
```

override说明符

- 检查是否符合重写规则

```
class Base {  
public:  
    virtual void some_func() {}  
    virtual void foo(int x) {}  
    virtual void bar() const {}  
    void baz() {}  
};  
class Derived : public Base {  
public:  
    virtual void sone_func() override {}  
    virtual void foo(int &x) override {}  
    virtual void bar() override {}  
    virtual void baz() override {}  
};
```

final说明符

- 阻止派生类去继承基类的虚函数

```
class Base {  
public:  
    virtual void foo(int x) {}  
};
```

```
class Derived : public Base {  
public:  
    void foo(int x) final {};  
};
```

```
class Derived2 : public Derived {  
public:  
    void foo(int x) {};  
};
```

final说明符

- 阻止类被作为基类

```
class Base final {  
public:  
    virtual void foo(int x) {}  
};
```

```
class Derived : public Base {  
public:  
    void foo(int x) {};  
};
```



感谢聆听
欢迎关注