

Lesson 11

- 1.面向对象的第二个特征是继承（inherit）。
- 2.在类的继承体系中，父类扮演的是一种更加泛化、抽象的角色，子类则是一种更加具体的角色。
- 3.在继承中，父类含有的数据成员，子类中全部都有。
- 4.总结三种访问权限：
 - a) private 成员只能在类的内部被访问
 - b) protected 成员在家族继承体系中是可见的
 - c) public 成员在任意地方都是可见的
- 5.private 也被子类继承，只是无法被直接访问。
- 6.通过子类对象来调用函数：
 - a) 从父类继承而来的函数，可以正常使用
 - b) 子类自行添加的函数，可以正常使用
 - c) 子类编写和父类重名的函数，不管是否改变了参数，调用的都是子类的版本。如果需要调用父类的版本，需要显式指定父类名
- 7.子类重定义父类的函数，不管有没有改变参数，当使用子类对象去调用这些函数的时候，调用的总是子类自定义的版本，我们称之为子类隐藏了父类的函数。
- 8.C++类的对象，只包含普通数据成员。
- 9.子类对象在构造的时候，会先生成一个父类的无名对象。

所以子类对象内部包含一个父类对象，因此，把子类对象赋值给父类对象是合法的，在赋值过程中，子类对象多余的部分被丢弃，这个赋值过程成为对象的切除。

10.类对象的构造顺序：

- a) 先构造父类
- b) 构造成员对象
- c) 调用自己的构造函数
- d) 析构顺序与之相反。

11.自己编写的 NonCopyable 要通过私有继承来使用，这是一个特例。

12.练习：写一个 Thread 类，然后派生出生产者和消费者线程。

13.基类的指针可以指向派生类的对象。但是使用该指针调用函数时，该指针把自己指向的派生类对象，当做基类对象。

14.用基类指针指向派生类对象，当使用该指针调用函数时：

- a) 基类中有，派生类继承过去的函数：调用的是基类版本
- b) 基类中没有，派生类自行添加的函数：无法调用。
- c) 派生类重写了基类的同名函数，此时通过基类指针调用的仍是基类的版本。

15.继承体系中有这样一个结论：派生类 is 基类。

- a) 从下到上转化（派生类的指针转化为基类指针）没有

任何问题。

b) 从上到下，需要程序员手工保证逻辑上的正确性。

16.继承体系中：

a) 从上到下是一个**具体化**的过程，这个过程有无数个选择，人可以转化成学生、工人、教师多种型别。

b) 从下到上是一个**抽象、泛化**的过程，这里只存在一种选择。

17.从上到下的转化成为**向下塑形**，本质上是不安全的，需要谨慎使用。

18.以函数重载为例，编译器在编译期间就可以确定调用重载的哪个版本，也就是**将函数调用与实际代码实现绑定**。这个过程是在**编译期间**完成的，所以称为**静态绑定（static binding）**，或者早绑定。

19.满足动态绑定的条件：

a) 用基类的指针或者引用指向派生类的对象

b) 通过该指针或引用，调用的函数为 virtual 函数

20.动态绑定就是：编译器在编译期间并不确定到底调用哪个函数，而是把函数调用和实际代码绑定的过程推迟到运行期间。

21.在 Animal 和 Cat、Dog 的示例中，当编译器检测到 `pa->display();`时，发现：

a) `pa` 为 Animal 类型指针

b) display 为 virtual 函数

于是编译器，停止绑定，而是等待到运行期，根据 pa 所实际指向的对象，来确定调用哪个函数。

21.使用 virtual 修饰的函数称为虚函数。虚函数具有继承性，只要子类的同名函数参数与基类相同，返回值兼容，那么子类的该函数也是虚函数。

22.虚函数与普通函数的区别：

- a) 普通函数总是采用静态绑定，根据指针类型来确定调用哪个函数。
- b) 虚函数在触发动态绑定的情况下，根据指针所指向的实际对象类型，来决定调用哪个函数。

23.下列情形：

- a) 基类中有 display 虚函数，派生类中也有 display，参数与 display 相同，返回值兼容，此时派生类的 display 也是虚函数，那么我们称派生类的 display 覆盖了基类的 display。
- b) 基类中 display，而且是 virtual 函数，但是派生类中的 display 不满足虚函数继承的条件（参数改变或者返回值不兼容），此时派生类中的 display 不是虚函数。
- c) 基类中的 display 不是虚函数，派生类中 display 此时不必关心形参和返回值得问题。
- d) 我们把后两者称为派生类函数隐藏了基类的 display 函

数。

24.只需要记住动态绑定的情形，也就是函数覆盖的情形，其他所有的均为函数的隐藏。

25.函数覆盖：指的是派生类的 virtual 函数覆盖了基类的 virtual 函数。通过基类指针调用 virtual 函数来体现。

26.函数隐藏：通过派生类对象调用自身改写的与基类同名的函数，此时调用的是自己改写的版本，所以称为隐藏。

27.函数重载称为一种编译期的多态，动态绑定称为运行时的多态，但是面向对象所指的多态（polymorphism），只包括动态绑定。

28.面向对象的第三个特征是动态绑定（Dynamic Binding）。

29.练习：Computer 派生出相应的品牌。Dell、Lenovo、Mac，实现 price()函数。