

1 多态的原理

2015年12月6日 8:46

```
class Parent
{
public:
    virtual void func(int a, int b)
    {
        cout << "Parent: func(inta ,int b) ..." << endl;
    }

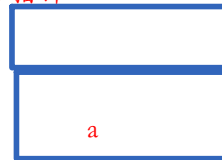
    virtual void func(int a)
    {
        ...
    }

    void func(int a, int b, int c)
    {
        cout<< "Parent: func(int a, int b , int c)" << endl;
    }
private:
    int a;
};
```

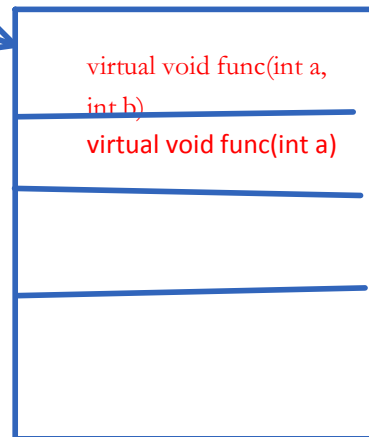
```
class Child :public Parent
{
public:
    virtual void func(int a, int b)
    {
        cout << "Child :func( int a , int b )..." << endl;
    }

private:
    int b;
};
```

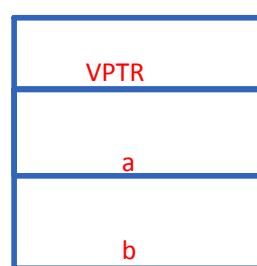
Parent p; // 定义一个父类对象
VPTR指针



Parent::虚函数表

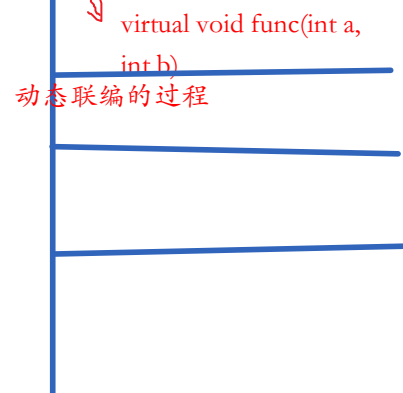


Child c; // 顶一个子类对象的时候



Parent*p = &c;
p->func(10, 20);
p->func(10,20,30)

Child::虚函数表



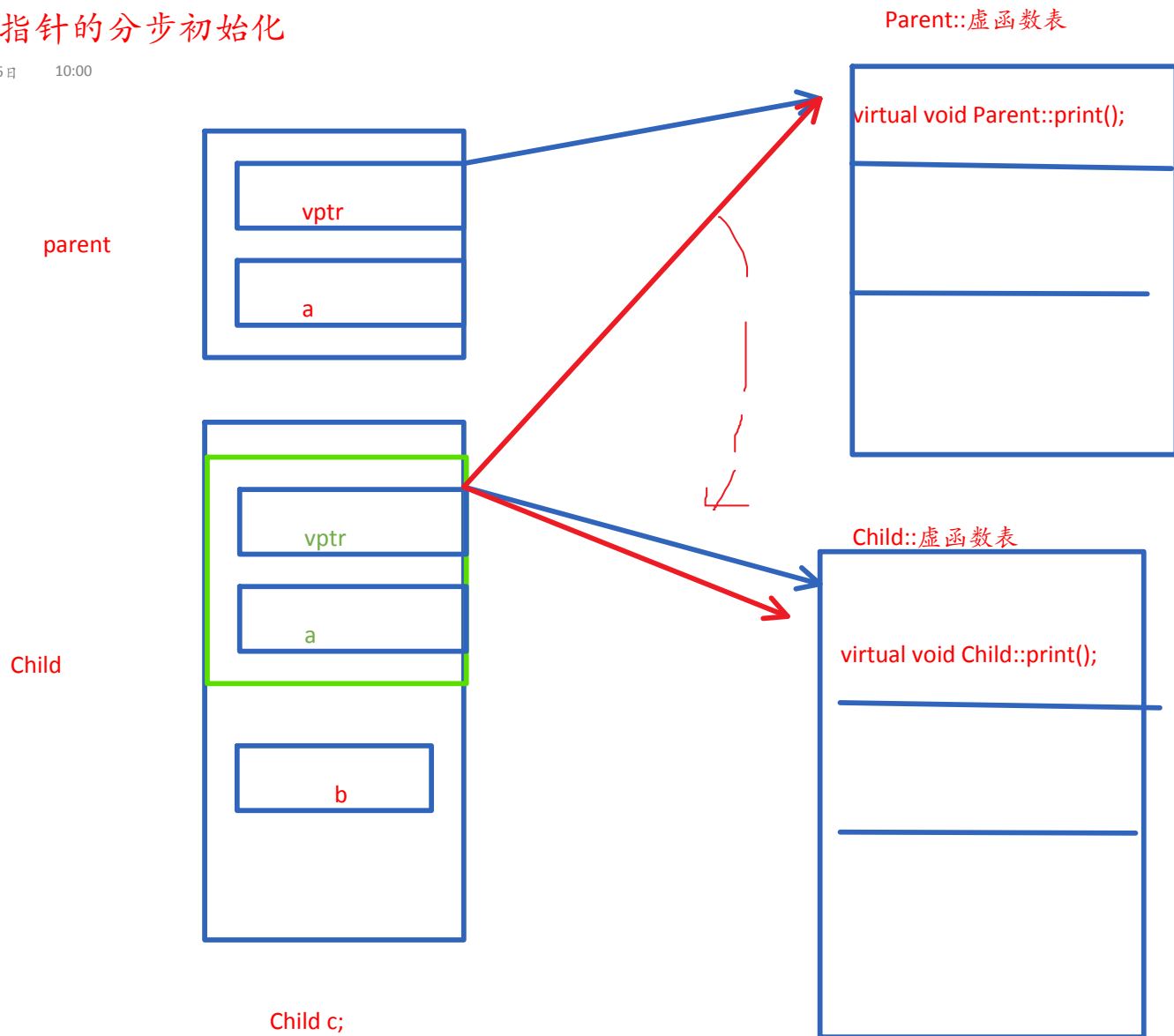
动态联编的过程

Parent *p = &c;

p->func(10, 20)

2 vptr指针的分步初始化

2015年12月6日 10:00

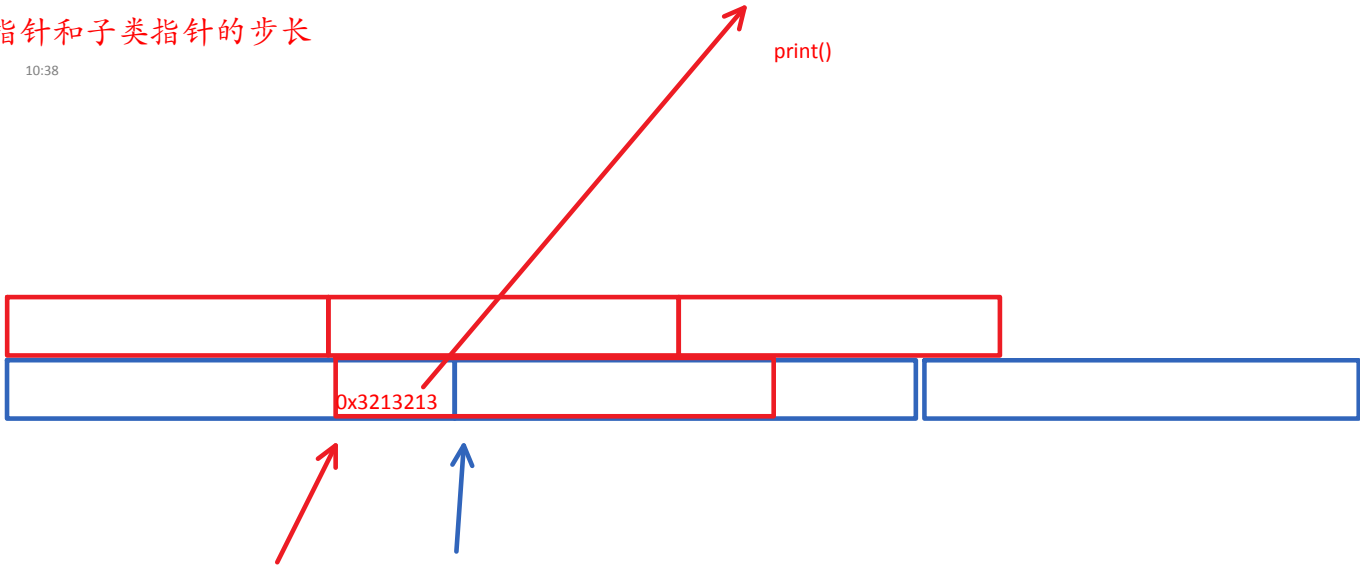


Child c;

- 1 先调用Parent的构造函数来构造
- 2 在调用父类的构造函数的时候，子类的vptr指针此时指向的是父类的虚函数表
3. 在调用完父类的构造函数之后，子类的vptr指针会执行子类的虚函数表。

3 父类指针和子类指针的步长

2015年12月6日 10:38

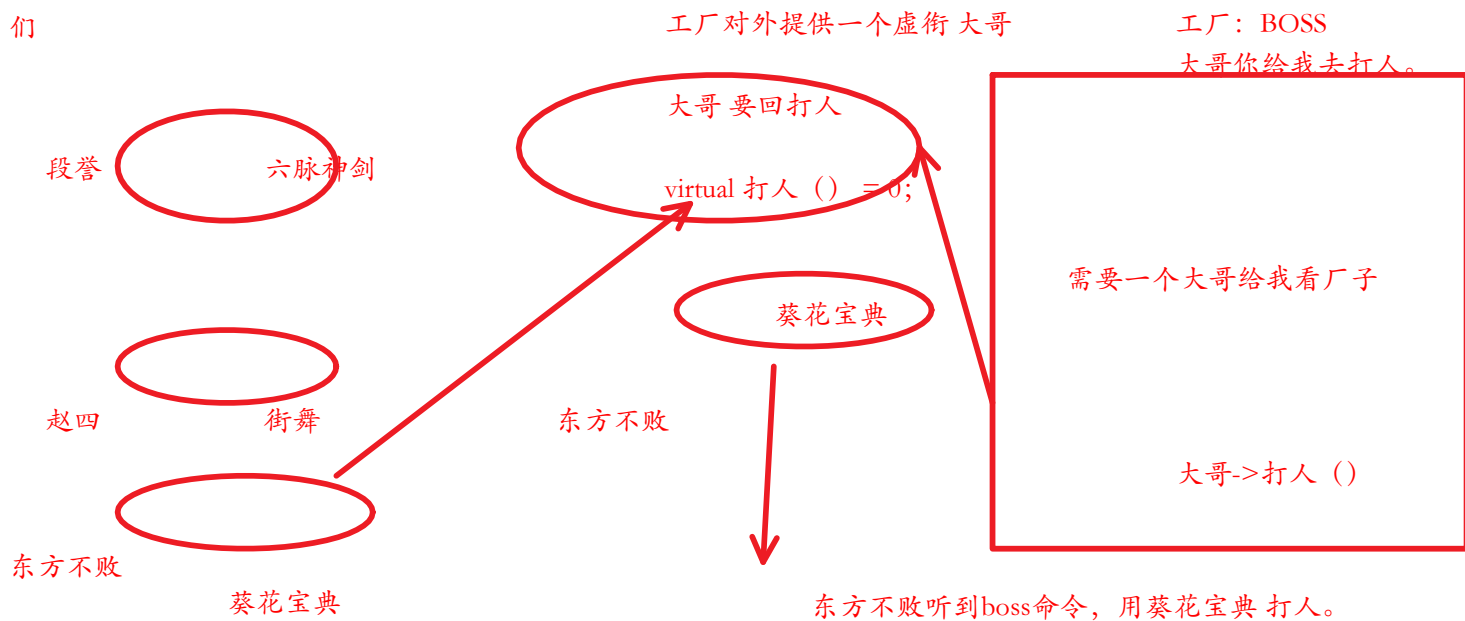


4 多态的案例

2015年12月6日 14:50

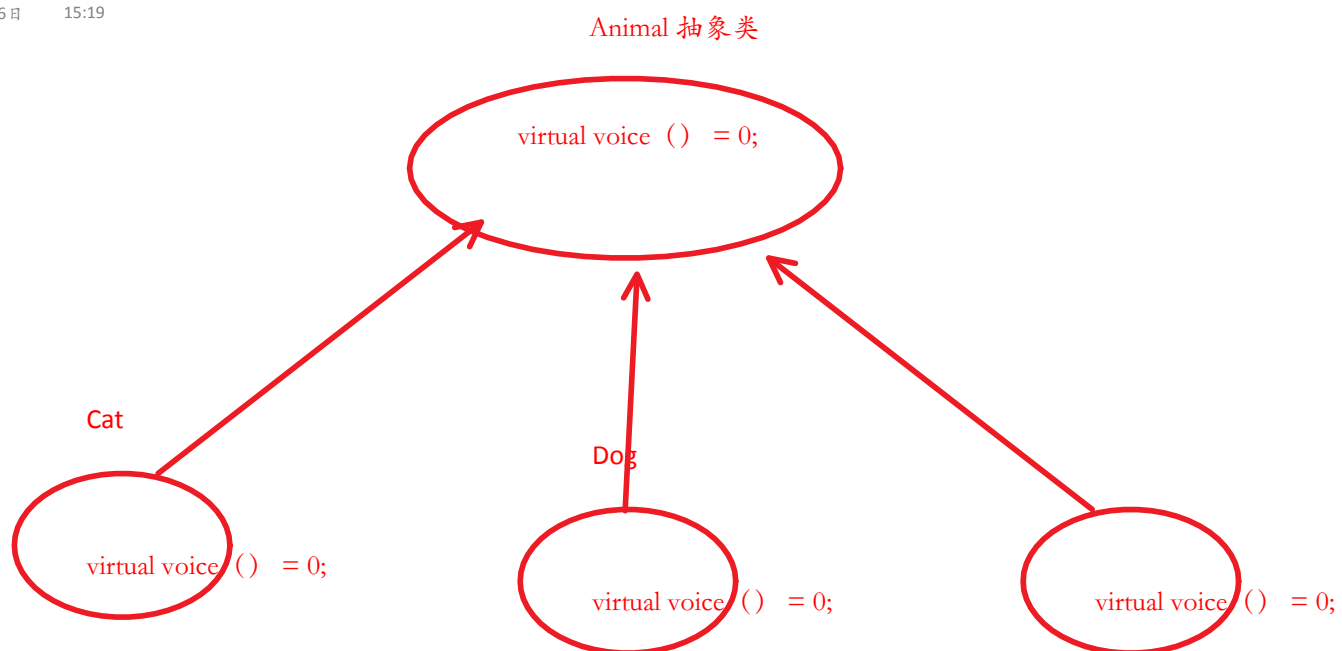
1. 要有继承 东方不败 继承了 大哥的虚街
2. 要有虚函数或者纯虚函数重写。 东方不败重写了打人方法
3. 父类指针指向子类对象。 大哥你给我去打人。

社会上出现了一些梦想成为大哥的男人们



5 动物园

2015年12月6日 15:19



6 电脑组装的案例

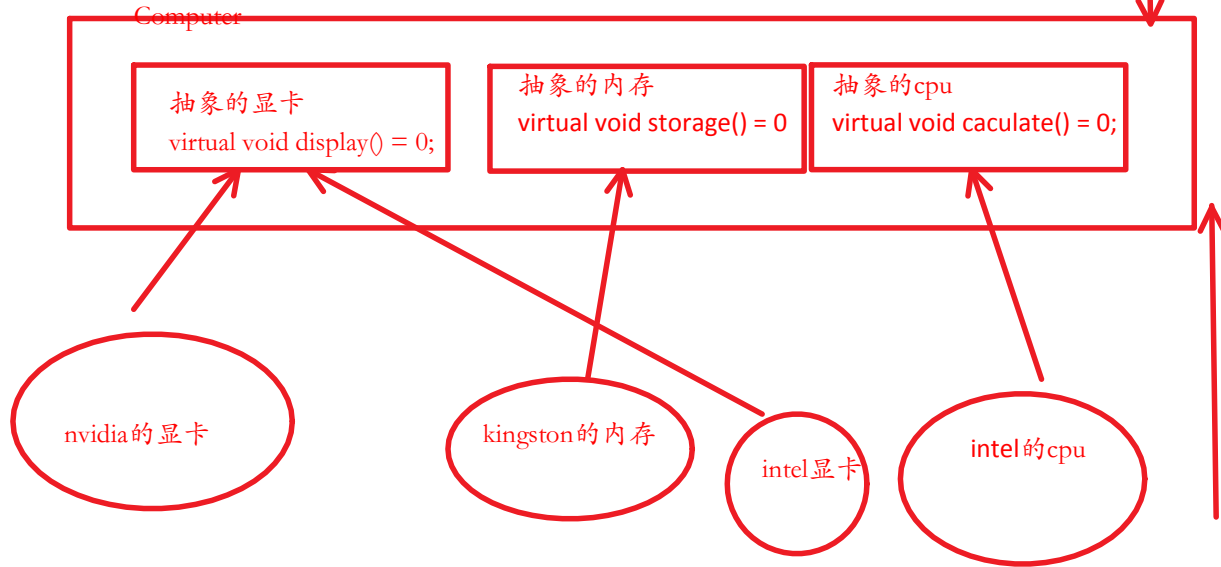
2015年12月6日 16:08

依赖倒转原则

main 高层业务逻辑层

要求组装两台电脑，1 -- 用intel系列的， intel的cpu intel的显卡 intel内存、
2 --- intel cpu nvidia 显卡 kingston的内存条

联想公司构造一个
抽象层



实现层
各个厂商去根据
抽象层来实现
自己的产品类