


```
{
public:
    A() {}
    virtual void foo()
    {
        cout<<"ThisisA."<<endl;
    }
};

classB:publicA
{
public:
    B() {}
    void foo()
    {
        cout<<"ThisisB."<<endl;
    }
};

int main(intargc, char*argv[])
{
    A* a = new B();
    a->foo();
    if(a != NULL)
        delete a;
    return0;
}
```

这将显示：

This is B.

如果把virtual去掉，将显示：

This is A.

前面的多态通过使用虚函数virtual void foo()来实现。

在java中：

多态，是面向对象的程序设计语言最核心的特征。多态，意味着一个对象有着多重特征，可以在特定的情况下，表现不同的状态，从而对应着不同的属性和方法。从程序设计的角度而言，多态可以这样来实现（以java语言为例）：

```
public interface Parent//父类接口
{
    public void simpleCall();
}

public class Child_A implements Parent
{
    public void simpleCall();
    {
        //具体的实现细节；
    }
}

public class Child_B implements Parent
{
    public void simpleCall();
    {
        //具体的实现细节；
    }
}
```

//当然还可以有其他的实现

然后，我们就可以看到多态所展示的特性了：

Parent pa = new Child_A();

pa.simpleCall()则显然是调用Child_A的方法；

Parent pa = new Child_B();

pa.simpleCall()则是在调用Child_B的方法。所以，我们对于抽象的父类或者接口给出了我们的具体实现后，pa可以完全不用管实现的细节，只访问我们定义的方法，就可以了。事实上，这就是多态所起的作用，可以实现控制反转。这在大量的J2EE轻量级框架中被用到，比如Spring的依赖注入机制。

- 1 什么是多态

2 web前端学习路

3 mvc教程

4 净水器厂家

5 linux培训

6 django 教程

7 塔钟

8 网页制作

9 日语学习

10 编程学习入门

11 spring 教程
- 12 高速摄像机

13 如何自学编程

14 艺术类院校

15 自考专升本

16 三级分销系统

17 ps培训

18 网站制作公司

19 网站建设

20 web前端培训

21 三室二厅装修效果图

22 电脑编程入门

分享

猜你喜欢

什么叫多态

广州驾校

肯德基宅急送网上订餐

祛痘产品排行榜10强

库存管理





grg是什么



电脑编程入门自学



单片机培训



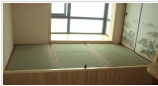
单片机培训机构



python视频教程



参数化



榻榻米床价格

新手上路

- 成长任务
- 编辑入门
- 编辑规则
- 百科术语

我有疑问

- 我要质疑
- 我要提问
- 参加讨论
- 意见反馈

投诉建议

- 举报不良信息
- 未通过词条申诉
- 投诉侵权信息
- 封禁查询与解封

分享

- ☆
- 👁
- 💬
- 🐼

百科美图编辑赛
等你参加