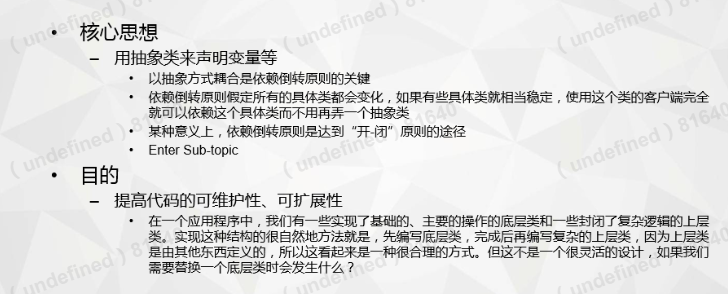
# 三、依赖倒置原则

2018.05.06 23:33:28字数 234阅读 143



### 1、定义

##### 依赖倒置原则，高层模块不应该依赖底层模块，两者都应该依赖其抽象；抽象不应该依赖细节；细节应该依赖抽象。

##### 

### 2、实现

### （1）定义车接口

public interface ICar {

void run();

}

### （2）定义司机接口

public interface IDriver {

void drive(ICar car);

}

### （3）司机实现类

public class Driver implements IDriver {

@Override

public void drive(ICar car) {

car.run();

}

}

### （4）奔驰车实现类

public class Benz implements ICar {

@Override

public void run() {

Log.d("JimmyZhang", "奔驰汽车开始运行...");

}

}

### （5）宝马车实现类

public class BMW implements ICar {

@Override

public void run() {

Log.d("JimmyZhang", "宝马汽车开始运行...");

}

}

### （7）测试实现

public class Client {

public static void main(String[] args) throws IOException {

IDriver zhangsan = new Driver();

ICar benz = new Benz();

zhangsan.drive(benz);

}

}

### 3、依赖的三种写法

##### （1）构造函数传递依赖对象；

##### （2）Setter方法传递依赖对象；

##### （3）接口声明依赖对象。

### 4、最佳实践

##### （1）每个类尽量都有接口或者抽象类，或者抽象类和接口两者都具备；

##### （2）变量的表面类型尽量是接口或者抽象类；

##### （3）任何类都不应该从具体类派生；

##### （4）尽量不要覆写基类的方法；

##### （5）结合里氏替换原则使用。