

## 1.编写一个Car接口, 提供run方法

```
public interface Car {  
    public void run();  
}
```

## 2.编写一个Falali类实现Car接口,重写run方法

```
public class Falali implements Car {  
    @Override  
    public void run() {  
        System.out.println("法拉利以每小时500公里的速度在奔跑.....");  
    }  
}
```

## 3.编写一个Benchi类实现Car接口

```
public class Benchi implements Car {  
    @Override  
    public void run() {  
        System.out.println("奔驰汽车以每秒1米的速度在挪动.....");  
    }  
}
```

## 4.提供一个CarFactory(汽车工厂),用于生产汽车对象

```
public class CarFactory {  
    /**  
     * @param id : 车的标识  
     *          benchi : 代表需要创建Benchi类对象  
     *          falali : 代表需要创建Falali类对象  
     *          如果传入的车标识不正确,代表当前工厂生成不了当前车对象,则返回null  
     * @return  
     */  
    public Car createCar(String id) {  
        if("falali".equals(id)) {  
            return new Falali();  
        } else if("benchi".equals(id)) {  
            return new Benchi();  
        }  
        return null;  
    }  
}
```

## 5.定义CarFactoryTest测试汽车工厂

```
public class CarFactoryTest {  
    public static void main(String[] args) {
```

```
    CarFactory carFactory = new CarFactory();  
    Car benchi = carFactory.createCar("benchi");  
    benchi.run();  
    Car falali = carFactory.createCar("falali");  
    falali.run();  
}  
}
```