工厂模式

定义一个用于创建产品的接口，由子类决定生产什么产品，工厂模式使其创建过程延迟到子类进行。

特点：模式的表现只是一个抽象的方法。提前定义用于创建对象的接口，让子类决定实例化具体的某一个类，即在工厂和产品中间增加接口，工厂不再负责产品的创建，由接口针对不同条件返回具体的类实例，由具体类实例去实现。工厂方法模式每个具体工厂类只完成单一任务，代码简洁。工厂方法模式完全满足OCP，即它有非常良好的扩展性。

public interface Phone {

    void make();

}

public class HuaWeiPhone implements Phone{

    @Override

    public void make() {

        System.out.println("make huawei phone");

    }

}

public class XiaoMiPhone implements Phone{

    @Override

    public void make() {

        System.out.println("make xiaomi phone");

    }

}

public class PhoneFactory {

    public Phone makePhone(String phoneType) {

        if(phoneType.equalsIgnoreCase("xiaomi")){

            return new XiaoMiPhone();

        }

        else if(phoneType.equalsIgnoreCase("huawei")) {

            return new HuaWeiPhone();

        }

        return null;

    }

}

public class Demo {

    public static void main(String[] args) {

        PhoneFactory phoneFactory = new PhoneFactory();

        Phone huawei = phoneFactory.makePhone("huawei");

        huawei.make();

        Phone xiaomi = phoneFactory.makePhone("xiaomi");

        xiaomi.make();

    }

}