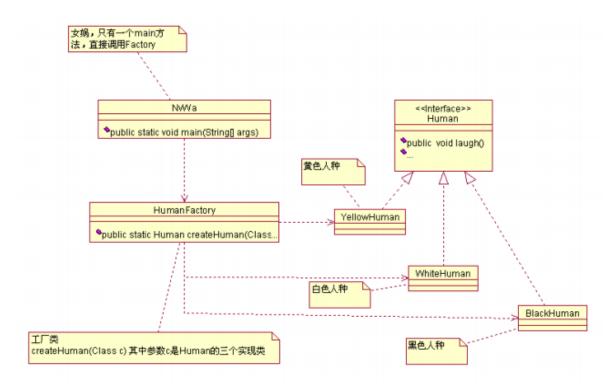


首先定义人,人的接口,和其实现类,创建工厂,可以生产各种类型的人,再用女娲来调用工厂类

```
定义人类的接口
public interface Humen(){
public void laugh();
public void cry();
public void do();
}
定义接口的实现类
public class Whilt implement Humen(){
//实现方法
}
public class black implement Humen(){
//实现方法
}
public class Factory(){
public static Humen createHumen(className){
```

```
Humen humen = null;
human = (Human)Class.forName(className).newInstance();
}

定义女娲
public class Maker(){
public static void main(String [] args){
Humen humen = Factory.createHumen(whilt.class);
}
}
```



两个八卦炉,一个造女的,一个造男的,开足马力,一直造到这个世界到现在这个模式为止。

抽象工厂模式讲完了,那我们再思考一些问题:工厂模式有哪些优缺点?先说优点,我这人一般先看人优点,非常重要的有点就是,工厂模式符合 0CP 原则,也就是开闭原则,怎么说呢,比如就性别的问题,这个世界上还存在双性人,是男也是女的人,那这个就是要在我们的产品族中增加一类产品,同时再增加一个工厂就可以解决这个问题,不需要我再来实现了吧,很简单的大家自己画下类图,然后实现下。

那还有没有其他好处呢?抽象工厂模式,还有一个非常大的有点,高内聚,低耦合,在一个较大的项目组,产品是由一批人定义开发的,但是提供其他成员访问的时候,只有工厂方法和产品的接口,也就是说只需要提供 Product Interface 和 Concrete Factory 就可以产生自己需要的对象和方法,Java 的高内聚低耦合的特性表现的一览无遗,哈哈。