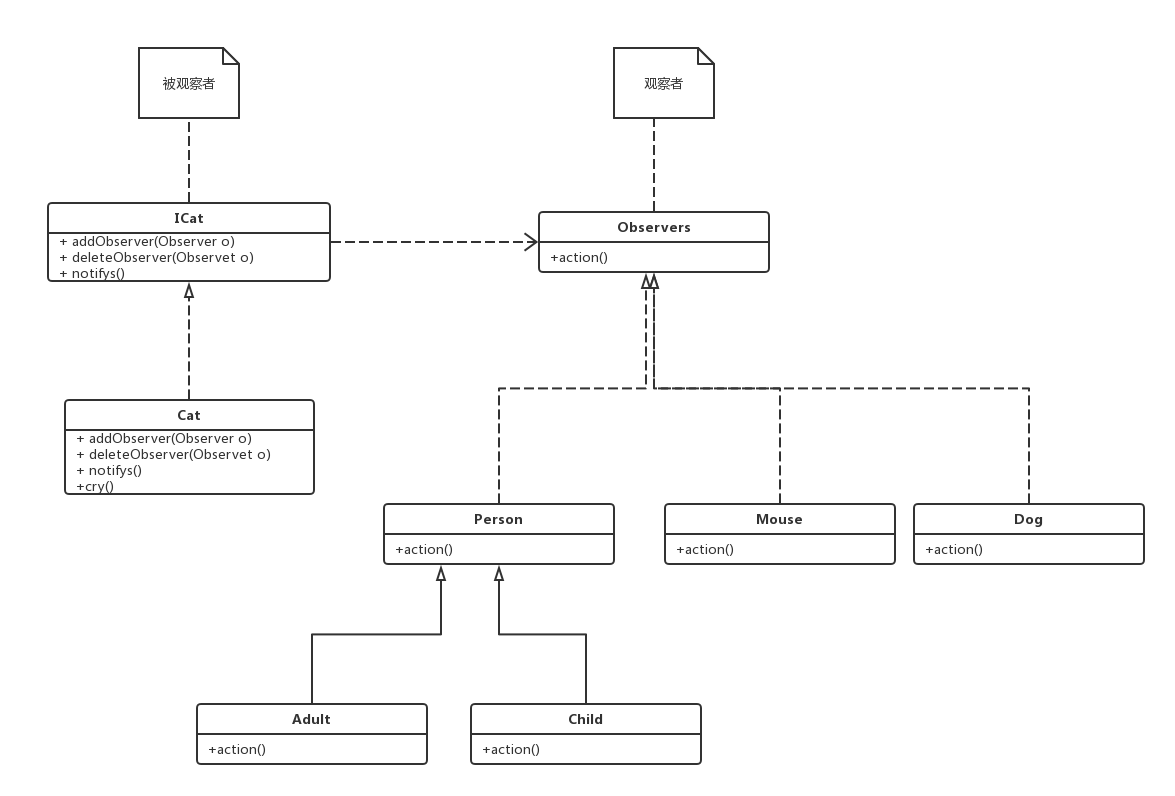
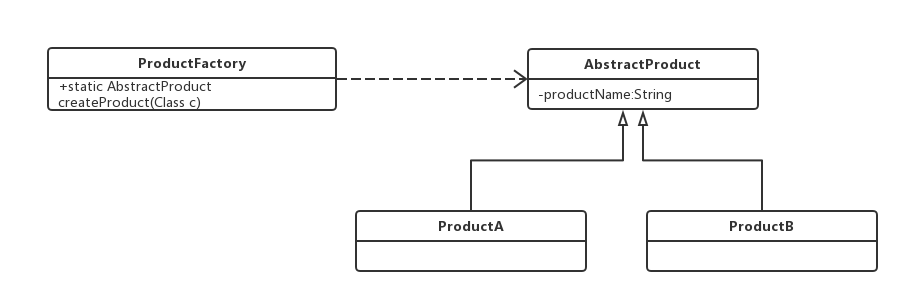
**设计 模式**

观察者模式：



静态工厂方法：



核心代码：

public class SimpleProductFactory {

public static <T extends AbstractProduct> T createProduct(Class c){

AbstractProduct ap = null;

try {

ap = (AbstractProduct) Class.forName(c.getName()).newInstance();

} catch (Exception e) {

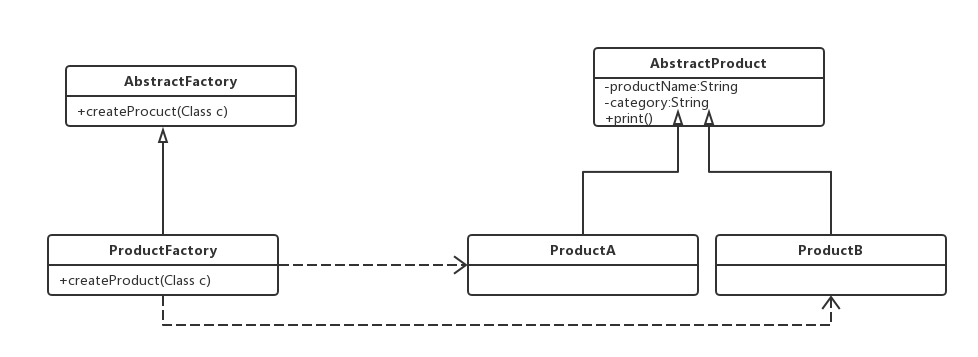
e.printStackTrace();

}

return (T) ap;

}

}

抽象工厂模式：

核心代码：

public abstract class AbstractFactory {

public abstract <T extends AbstractProduct> T createProduct(Class c);

}

实现类工厂：

public class ProductFactory extends AbstractFactory {

@Override

public <T extends AbstractProduct> T createProduct(Class c) {

AbstractProduct ap;

try{

ap = (AbstractProduct) Class.forName(c.getName()).newInstance();

return (T) ap;

}catch(Exception e){

e.printStackTrace();

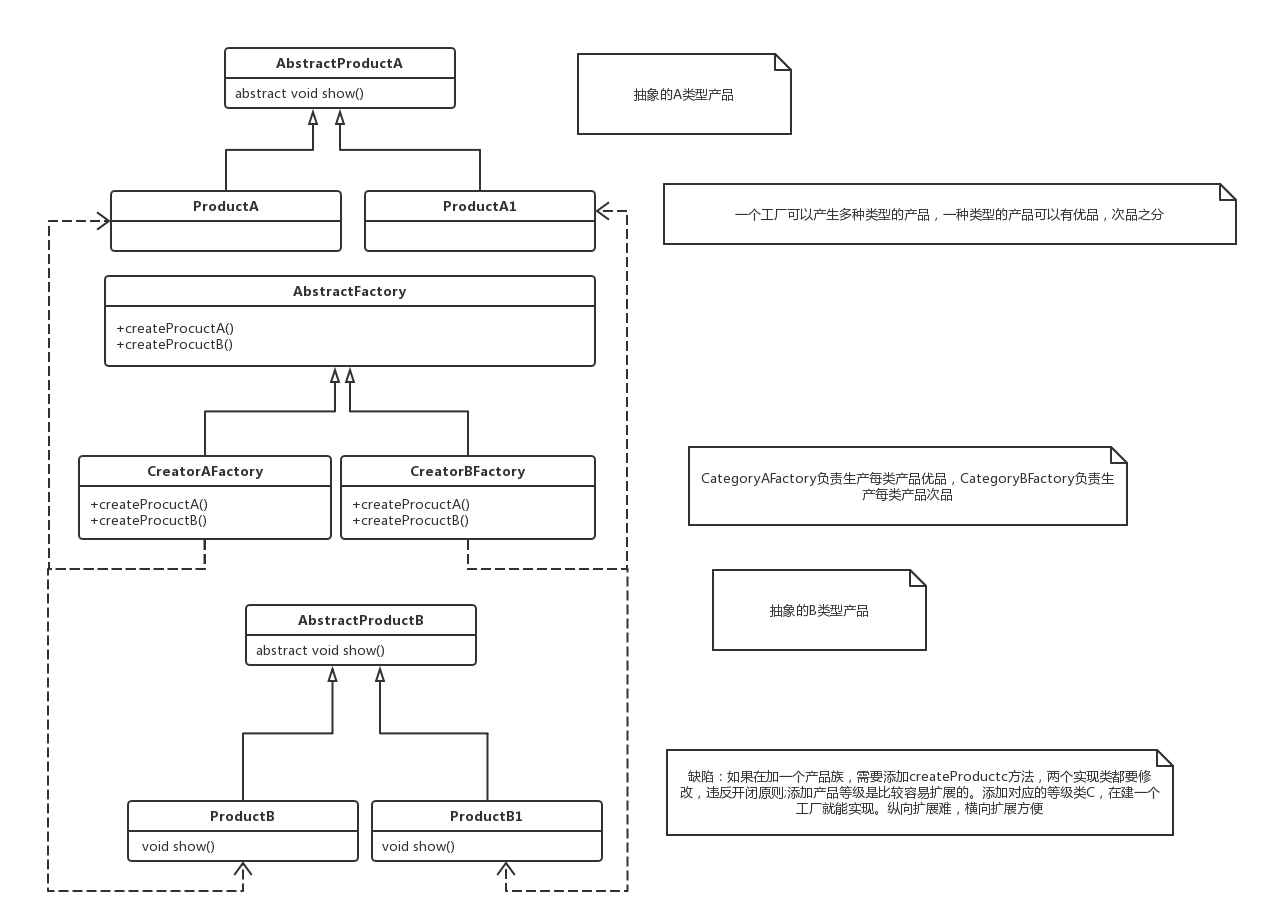
}

return null;

}

}

抽象工厂：



策略设计模式：

