工厂可以是工厂类，也可以是工厂方法

// 工厂类

class ProductFactory{

    public Product createProductA(){

        ....

    }

}

// 工程方法

class Product{

    public static Product createProductA(){

        ....

    }

}

工厂负责创建复杂的类或聚合

工厂的每个操作是原子的，要么成功，要么失败

如果工厂接收一个创建请求但无法完成创建，应该抛出异常

聚合的工厂方法一般放置在聚合根，也可以将一个聚合的工厂方法放置在另一个聚合内部

**工厂类创建原则**

创建满足最小规则的聚合，其他选择留给外部，如

创建User只需要用户名和密码即可，其他属性（如地址，电话等非必须属性）留给外部

class Product{

    // name 在生成后不允许改变，所以只有 get

    public string Name { get; }

public string DisplayName { get; set; }

    protected Product(string name){

        Name = name;

    }

    // name 是必须的，所以只需要 name

    public static Product createProductA(){

        // 这里为了方便，但方法内直接使用"\*"字符是不推荐的，怎么重构？请阅读重构

        return new Product("A");

    }

}

**有些时候，直接使用构造函数是最佳选择**

构造函数应该保持简单

构造函数中不要调用其它类的构造函数，那怎么办？

ClassA必须有ClassB那么， new ClassA(classB)