设计一个重要的功能就是抽象，我们使用的面向对象语言class都会继承object，为什么会继承object，因为object就是所有事物的抽象。同理，四色原型就是所有设计的抽象

如下2个问题：

1. 线程实例一个类
2. 用户下单

面对这2个描述，你是如何给出对应的设计呢？请看下面的4色原型

**什么是四色原型**

四色原型将所有的对象分为了4种类型，四色原型的表述：

“什么东西通过什么方式参与到什么操作中”

基本上我们人类的所有活动都可以用这段抽象来表达，下面介绍这4种抽象原型

**moment-interval archetype（MI，时刻原型）**

MI表示在某个时刻发生的操作，因为业务要求或合法性原因需要跟踪，如下示例：

卖东西是在某个时刻发生的，它有发生日期和时间。租赁行为是在一段时间内发生，从开始出租和归还所租物品；预定也是持续一段时间，什么时候预定；什么时候过期等

**role archetype（角色原型）**

我们上面说MI表示在某个时刻发生的操作，那么是谁做的这个操作，我们的role原型就代表这个角色，如下：

外卖员送外卖，维修员修电梯，外卖员、维修员都是角色

**party, place, or thing archetype（PPT，事物）**

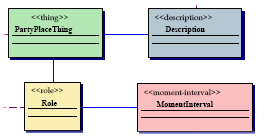
公司的员工有送外卖的，也有修电梯的，当员工送外卖时他具有外卖员role，当员工修电梯时他具有维修员role，role随员工的MI变化而变化，但员工这个是不会变的，这个示例中，员工就是PPT

PPT代表一个事物，更具体点我们可以认为是可以产生操作的事物（笔者个人理解）

**description archetype（描述）**

description是对PPT的一个描述，PPT所属的类别（如插座属于家用电器，员工A的现住地址XXX）

**如下四色原型的关系**



**最初的问题**

1. 线程实例类

根据4色原型得到的role是线程，MI是实例类

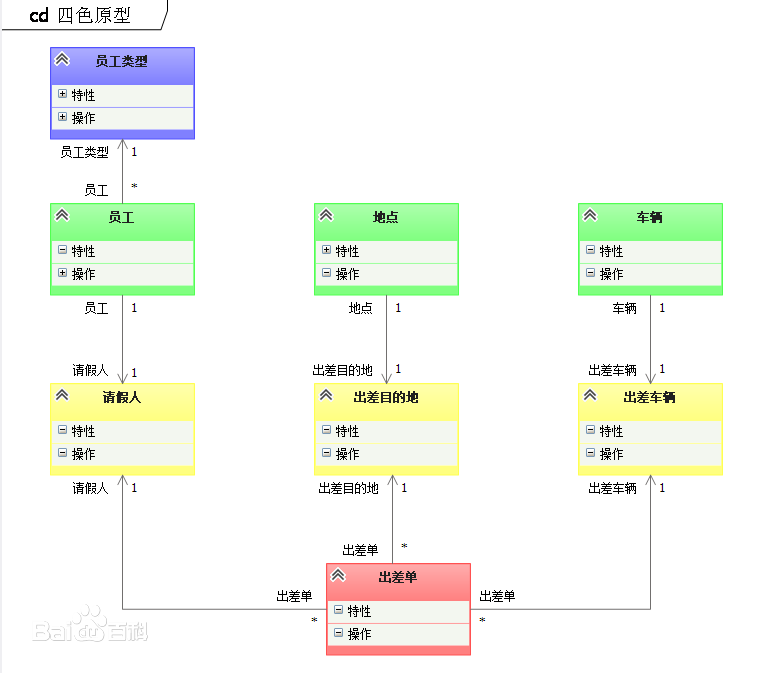
1. 用户下单

根据4色原型得到的role是用户，MI是下单

**必须的原型**

在一段描述中PPT，Role，MI是必须的，description为非必须的

**示例**



**测试**

外卖员将外卖送到客户家里

1. 人（PPT），外卖员（Role），送外卖（MI）
2. 地点（PPT），客户家（Role），送外卖（MI）