1. 如何做到用户点击某个电影后，上升该电影的热门度，并且在考虑高并发的情况下不影响性能？(难点)

采用消息队列，也就是说，用户点击某个电影后，传递一个消息，每传递一个消息就提升一次热门度，并将提升的热门度的值添加到缓存中。并且设定一个定时方法，每隔一段时间将缓存中的提升的热门度的值添加到数据库中。

1. 完成场次功能的查询，并自动删除过期的场次，保证所有的场次都是最新的三天（难点）

查询大于当前时间小于三天的所有场次。（是否能用缓存技术？因为数据是实时更新的，如果用缓存的话，就得对缓存中对象的时间进行判断了，这里还能使用二次查询吗？会不会使业务逻辑代码亢长）

1. 用户点击为占座的座位并点击提交时生成订单，并在15分钟内为用户占座，15分钟还未支付或者用户自己手动删除订单后占座取消（难点,考虑高并发）

用户查询所有座位情况时，第一次从数据库中查询到数据后将所有座位情况提交到缓存中，用户点击座位并提交时，生成订单，生成订单后修改缓存中该座位的占座情况，并同时也要对数据库进行修改（考虑到并发问题，这里应该使用锁机制，可能存在多个用户同时点击该座位并对这个座位进行重复修改，考虑到性能问题，是直接在代码层面上使用同步还是采用乐观锁的形式），并设定定时方法，15分钟后自动删除订单并且重新修改座位情况为未占座情况

可用技术点：延迟加载，消息推送（go easy框架）