[https://mp.weixin.qq.com/s?\_\_biz=MzA5MTkxMDQ4MQ==&mid=2648933201&idx=1&sn=b21aeda79e6e6a825826f08fef14f09e&chksm=88621b6fbf159279a2d9e3f195e1be888a9e20cdf95a637385fbd69b5e4be1a99c193da5a611&token=2027319240&lang=zh\_CN&scene=21#wechat\_redirect](https://mp.weixin.qq.com/s?__biz=MzA5MTkxMDQ4MQ==&mid=2648933201&idx=1&sn=b21aeda79e6e6a825826f08fef14f09e&chksm=88621b6fbf159279a2d9e3f195e1be888a9e20cdf95a637385fbd69b5e4be1a99c193da5a611&token=2027319240&lang=zh_CN&scene=21" \l "wechat_redirect)

## 案例讲解

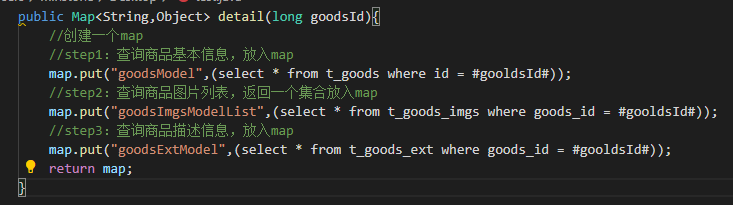
电商app都有用过吧，商品详情页，需要给他们提供一个接口获取商品相关信息：

* 商品基本信息（名称、价格、库存、会员价格等）
* 商品图片列表
* 商品描述信息（描述信息一般是由富文本编辑的大文本信息）

数据库中我们用了3张表存储上面的信息：

* 商品基本信息表：t\_goods（字段：id【商品id】、名称、价格、库存、会员价格等）
* 商品图片信息表：t\_goods\_imgs（字段：id、goods\_id【商品id】、图片路径），一个商品会有多张图片
* 商品描述信息表：t\_goods\_ext（字段：id，goods\_id【商品id】、商品描述信息【大字段】）

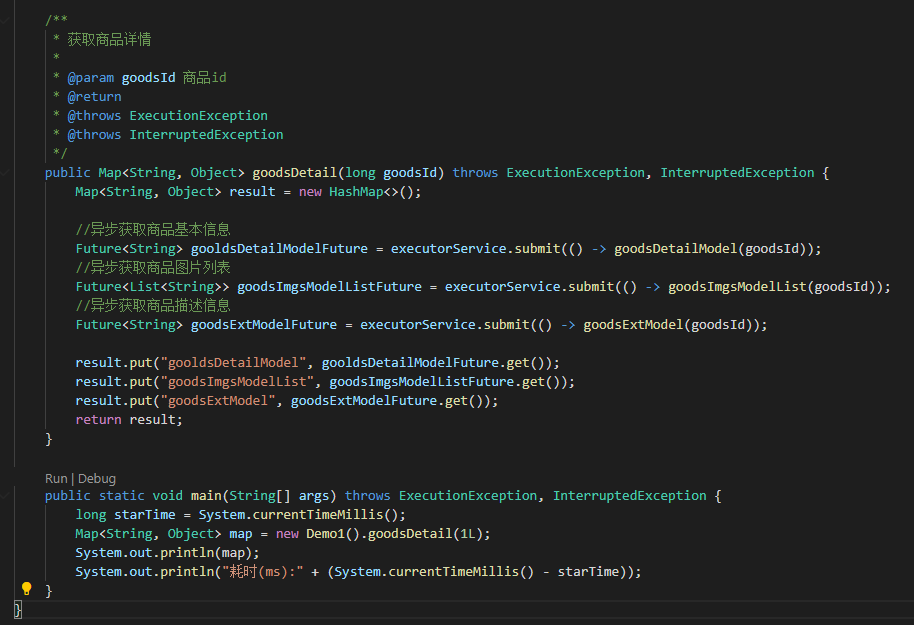
这需求对于大家来说很简单吧，伪代码如下：



上面这种写法应该很常见，代码很简单，假设上面每个步骤耗时200ms，此接口总共耗时>=600毫秒，其他还涉及到网络传输耗时，估计总共会在700ms左右，此接口有没有优化的空间，性能能够提升多少？我们一起来挑战一下。

在看一下上面的逻辑，整个过程是按顺序执行的，实际上3个查询之间是没有任何依赖关系，所以说3个查询可以同时执行，那我们对这3个步骤采用多线程并行执行，看一下最后什么情况，代码如下：





输出：



可以看出耗时200毫秒左右，性能提升了2倍，假如这个接口中还存在其他无依赖的操作，性能提升将更加显著，上面使用了线程池并行去执行3次查询的任务，最后通过Future获取异步执行结果。

**整个优化过程：**

* 先列出无依赖的一些操作
* 将这些操作改为并行的方式

**用到的技术有：**

* 线程池相关知识
* Executors、Future相关知识