

极客时间 Java 进阶训练营

第 29 课

业务系统重构--分享一个真实的重构案例



KimmKing

Apache Dubbo/ShardingSphere PMC

个人介绍

Apache Dubbo/ShardingSphere PMC

前某集团高级技术总监/阿里架构师/某银行北京研发中心负责人

阿里云 MVP、腾讯 TVP、TGO 会员

10 多年研发管理和架构经验

熟悉海量并发低延迟交易系统的设计实现

目录

1. A系统的现状和问题
2. 如何推动重构 A 系统
3. 重构的目标和方式
4. 重构的过程和结果
5. 对于本次重构的复盘
6. 总结回顾与作业实践

1. A 系统的问题与现状

A 系统的背景

业务上：从杭州刚挪到北京，作为重新孵化项目，想要做大做强。行业竞争压力非常之大，做不出成绩，就是生与死的问题。

技术上：遗留的老系统，再一起其他系统的基础上改的单体，团队拥抱变化，换手了4次以上，找不到一个做过这个系统的人了。

关系上：北京新业务定位是新业务的创新和孵化，这是一个大项目，也是要树立标杆的项目。

A系统的问题

当时我写的一封邮件，说的相对委婉：

国内机票旧系统开发于2006年，期间几十名开发人员在代码里留下了痕迹。不久前移交到北京新业务研发团队维护，主要问题有：

- 1、代码量大(40万行)，质量不高，结构复杂，难以修改和维护。一些代码的具体作用和逻辑细节，没有人清楚。很多注释和代码不一致，文档缺乏，比较混乱。
- 2、业务处理过程逻辑复杂，中间数据冗余，与外部接口耦合度高，性能较差，稳定性不高，扛不住大促活动压力。
- 3、对业务过程的数据和状态监控不足，难以管理控制和跟踪，各方使用不方便。
- 4、刚交接过来，没有人真正懂这个系统的业务，业务方也是空降，这个问题尤其严重。

重构面临的压力和选择

其实当时看来，重做是最好的路线，但不现实：

业务方：不能停止业务发展，死路一条

开发方：没有太多资源，也经不起折腾，半年后大促

2. 如何推动重构 A 系统

核心矛盾点在哪里？

这是一个很常见的矛盾点：

业务不认可重做/重构，伤筋动骨。

一提就说：反正系统还能用。

重做周期太长，成本太大。

重构不带来业务价值，步子大了也会影响业务。

祖传屎山，维护不易。

对于业务方

业务上：

必须要稳定推动业务发展，快速加功能，开发营销促销系统，不然被对手越拉越大。

核心要求：

- 1、业务不停
- 2、保障稳定
- 3、提升性能
- 4、快速开发

对于产品方

产品上：

提升用户体验，跟竞品在产品层面拉齐，多对接航司和第三方，引流，打通各方。

核心点：

- 1、用户体验
- 2、改版引流
- 3、同业务方

对于开发方

技术上：

最大问题是，资源不够。

最少的资源，最大满足业务和产品需求，同时能够发展北京研发团队。

最终达成一致

各方妥协，但是研发肯定是压力最大的。

不用怕，临危受命的同时，可以提要求，定规则。

不停机，发展业务，改进体验，都没问题。

本着这个大目标，要求按我们的安排来。

- 1、所有其他项目为这个项目让路，优先级 P0，（后来改成-1）
- 2、资源主要用来保障这个项目，所有人配合，包括后面招聘的
- 3、人员管理等方面，特事特办，所有人不管考勤等制度

3. 重构的目标和方式

怎么定重构目标？

平衡短期利益和长期利益。

分阶段，先重构搜索模块，然后是订单交易、产品库存。

国内机票搜索重构启动于6月底，经过近两个月的紧张开发，于8月底完成开发并提测，目前进入测试和准备上线阶段。

预计上线后，能带来更高的性能（+30%以上），更好的用户体验（数据的准确性、缓存策略），更好的代码质量与扩展性（更易于日常维护和添加功能），更好的监控和管理（at-eye）。

为国内机票年底的双12大促（预期投入一亿费用），做好最充足最全面的准备。

怎么定重构目标？

国内机票搜索重构的目标主要有五个：

- 1、**培养机票新人**。梳理清楚国内机票搜索部分的业务和各种潜规则，培养懂机票的新人。
- 2、**提高代码质量**。设计更合理，结构更清晰更易维护，系统耦合更低、扩展性更高。
- 3、**提升搜索性能**。使系统更稳定，更高效，搜索更快，资源消耗更低，大促不宕机。
- 4、**监控搜索过程**。统一管理搜索过程中的各种开关和参数调整，管理缓存，跟踪搜索步骤。
- 5、**国内机票搜索去 oracle 化**。在搜索重构的过程中，用 mysql 代替旧系统的 oracle 数据库。

目标的具体说明

- ① 培养新人问题，主要是XX\XX\XXX\XXXX。在此过程中，熟悉淘宝技术体系和开发过程，掌握国内机票业务，能逐步的 cover 机票系统。
- ② 代码质量问题，主要是项目结构合理的分层，搜索的分段式处理，code review，高覆盖率的单元测试，使项目易懂易上手易维护。
- ③ 提升性能方面，主要从简化中间结果和数据结构，理清业务简化旧逻辑，整理各步骤缓存最大化合理利用缓存结果三个方向来提升系统在 CPU 计算和内存、以及网络 IO 等方面的开销。提高系统吞吐量、QPS，降低响应时间。
- ④ 监控搜索过程，通过哈勃、timetunnel+ateye、memlog、系统 log 等手段埋点，管理系统多种粒度的开关和参数设置，跟踪和调试搜索流程中的数据和状态、各种外部接口数据和缓存。
- ⑤ 去 Oracle，改用 mysql+tdcl，实现分库分表，为进一步的分布式架构打下基础。

重构的组织方式

三个要点：

1. 小黑屋主力，核心+全流程
2. 其他后备力量，业务模块+迭代周期
3. 开发期间，试行996(全部算倒休)，抢在年底大促前搞定三期重构

4. 重构的过程和结果

重构怎么做--规划、启动

确定规划，分步改造

启动项目，拉齐思想



抄送: 淘宝-淘宝旅行; 淘宝-旅行技术; 淘宝-技术研发部-新业务技术团队

主题: 国内机票搜索重构项目: 夺天之战【涅槃】

天柱山，玄玄大陆第一高山。相传山高万仞，横亘在天地之间，绵延数十万里，直插入青天。故老相传，此山为天之尽头！天柱山，山连天，神魔止步不可攀！五千年前，天柱山塌陷，大批异世界敌人在此进入人类世界。

人界三大圣地遁世仙宫、至尊金城、梦幻血海，以及玄兽圣地天罚森林，集合所有强者聚于此地，鼎力合作，与异族一战，封邪魔于异界，大胜而归。

此战名曰“**夺天之战**”。

燕子坞，淘宝仙境第一名胜。据说由数千宛如珍珠般得珊瑚岛组成，蓝天白云，椰林树影，水清沙幼，风景如画，水天如诗。公认为天上地下最美之地。燕子坞，岛如珠，神佛留恋不复出！

奋战于此，

变革国内机票搜索系统架构，根治五年以来之沉痾，解耦系统，梳理业务，订正逻辑，精简数据。名为重构，实为重写。六人争分夺秒，日夜奋战，精诚合作，众志成城，呕心沥血，最终于近日功成。

破而后立，浴火重生，复建天界秩序；

勇于前行，乘风再起，中兴淘宝旅行。

史记称之为“**涅槃**”！！

Milestones

M1 上线启动10-11

M2 灰度上线10-20

M3 功能发布10-27

M4 全部提测11-08

M5 所有上线11-29

M6 机票大促12-21

重构怎么做--明确业务

1、重建业务需求文档。

需求是一切的起始点。

2、如何保障业务不停？

架构层面的设计，与产品方达成一致。

重构怎么做--架构调整

分层改造和逐步对接，借鉴与微软的 outlook 团队的案例。

这在当时是一个很大的冒险。

例如，页面的改版需求，我们顺带着把一些后端的业务逻辑，代码结构，数据库都改掉了。

在一个需求周期内，搭车业务需求，做技术改造。

重构怎么做--性能优化

从优化业务处理角度。同步大事务到小事务，异步处理。

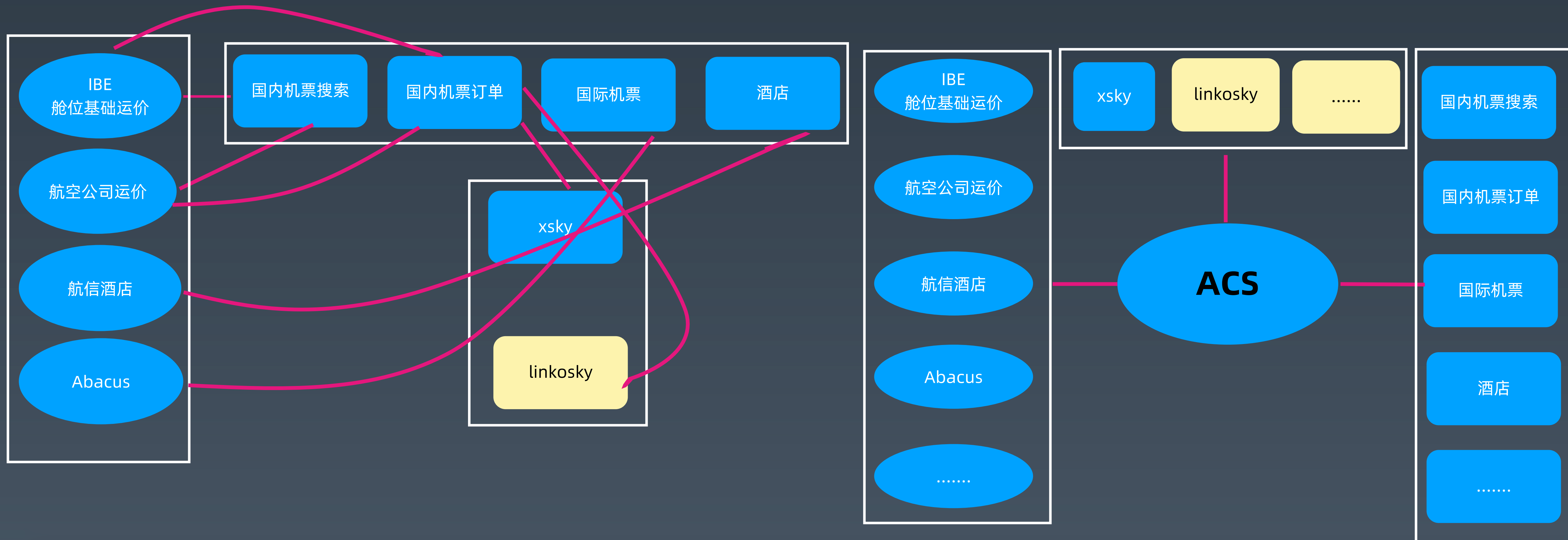
从优化系统代码角度。一个8层 for 循环的例子。

从优化数据结构角度。笛卡尔积的例子，一次请求500M内存。

从优化缓存策略角度，从静态有效期到动态有效期。

重构怎么做--引入标准

借鉴之前的 ESB 经验，解决对接的复杂性问题，**标准化集成方式**。
不同航司的接口，差异非常大，数据结构不同，数据的有效性不同。



重构怎么做--灰度发布

通过灰度开发+特性开关，保障可靠上线。

先是预发环境上线，办公室内访问，
然后线上灰度，公司内部IP访问，
再接着杭州和北京访问，
最后全国可以访问，完全放开。

重构怎么做--监控运维

实现一套全方位的运维系统，快速发现和处理问题。

实现线上开关和 debug 的机制，快速定位问题。

对每天的业务数据进行自动统计和发邮件，做到所有人对业务情况了然于心。

项目的效果

业务上，支持了大量新功能，包括营销促销系统。

技术上，重构了新系统，完成了去 O 和分库分表，保证了大促。

产品上，优化了用户体验，补齐了系统的各项短板。

团队上，培养了一批懂业务，懂新系统底层的研发核心人员。

新系统文档完善，架构先进，代码只有旧系统的1/3不到，性能提升30倍。

> 做减法比加法难。

此外，在新架构的基础上，打通了各个业务线。

后来我又负责了其他系统的重构，旅行线业务统一项目，关联销售项目等。

5. 对本次重构的复盘

复盘总结

为什么要做重构？

重构的难点是什么？

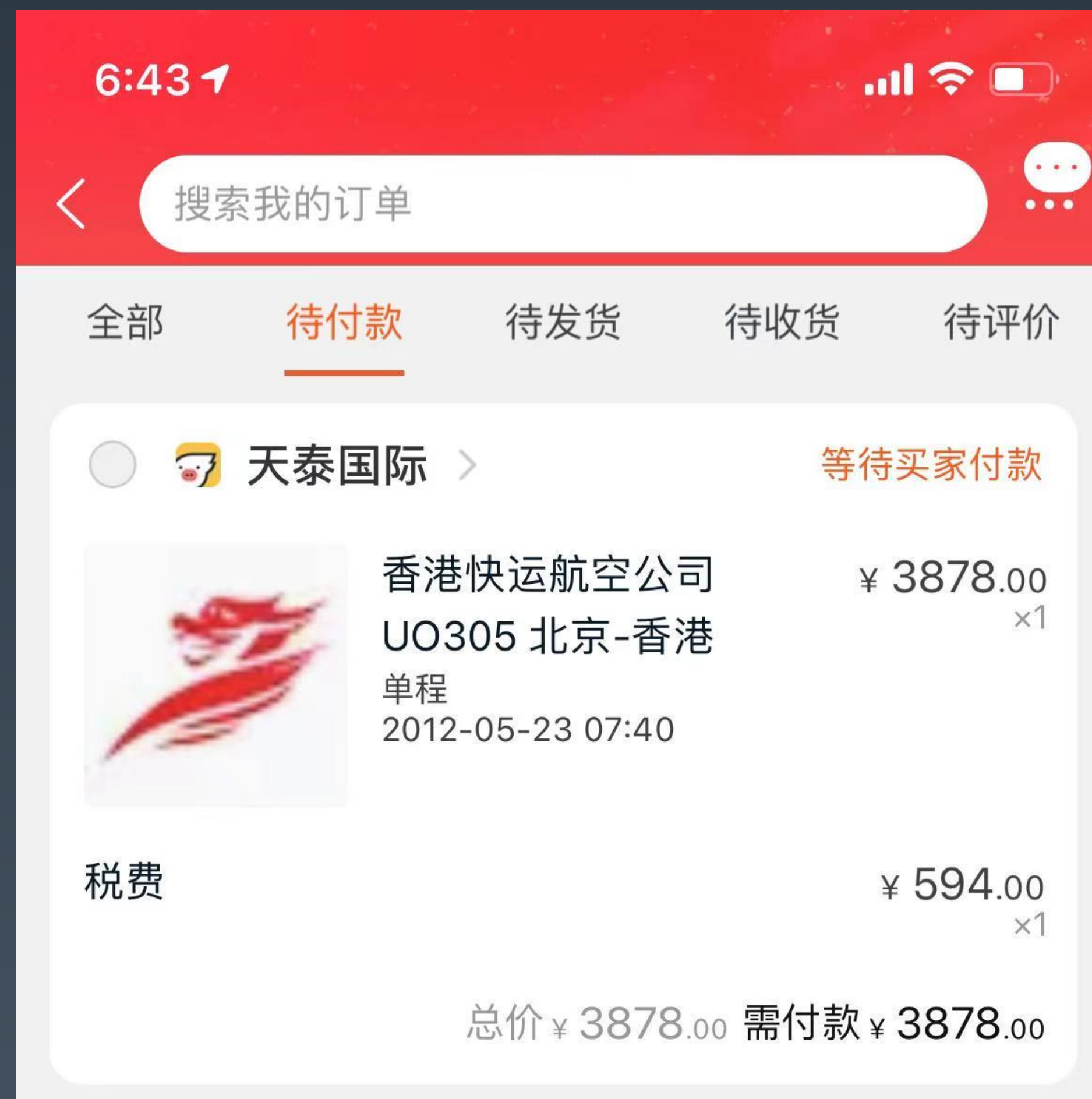
重构要做哪些事儿？

怎么解决重构过程的各种问题？

怎么保证重大改造的稳定上线？

一个彩蛋

机票的数据迁移。



6.总结回顾与作业实践

第 29 课总结回顾

项目的现状和问题

如何推动项目重构

重构具体要怎么做

重构的结果和思考

第 29 课作业实践

- 1、（选做）学习《[重构--改善既有代码的设计](#)》这本书，写读书笔记。
- 2、（选做）对于目前自己维护的项目代码，思考如何改善设计和实现。