最后的总结

最后一节课上吴老师说了一句话让我印象深刻，“如果把我们放在和你们同一个时代竞争，我们连汤都喝不上，你们已经比很多工程师要厉害了，只是少了一点对工程具体代码的熟练度而已。”

这样的一句话突然就让我释怀了，一个学期无数次熬夜通宵，似乎都有了理由。也突然明白窗户纸一旦破了，我们的行业有时没有那么高的技术门槛，有的部分需要的更多是学习的热情和抗压能力。也明白了，即使是经过我们一个学期的高强度训练，JAVA我们也只能说是摸到了一整套工程开发的门沿，我们进来了，进来了围城的第一重。站在门口，目力所及的还有重重叠叠的山，遥不可见的还有波波漾漾的海。我们所以为的高难度课程，原来也真的只是这个时代情况下的“通识教育”难度而已。

且回想自己这前15次作业吧。

（前半部分是对设计质量的回顾和思考）

第一周还没怎么看完Head First Java这种兴趣入门书的我，突然得知竟然以后每周都要写一个大作业一样的程序，而大作业需要用到的技术包含了无数看不懂的名词（不知道第一次作业有几个大佬一开始就知道正则表达式）。我麻痹自己，“没事的没事的，既然布置了就说明没那么难”。于是第一个周末在痛苦地摸索正则表达式中度过了。现在依然记得周一晚上终于完成了读取部分的代码，然后紧接着巧妙地将所有多项式全部读取进来用带有正负号的项来存储在链表中，之后根据系数大小快排，最后循环一遍合并相同系数项，输出即可得到计算结果。（回想起来这应该是最快得到优雅的设计方案的一次了）。过程相当激动人心，这也是第一次体会到课程的乐趣，也是最后一次最完整的乐趣（不夹杂痛苦的）。后来，我养成了写代码前仔细阅读指导书进行设计，脑内疯狂脑补的习惯，有时候想设计思路需要一两个小时，写在草稿纸上删删改改，这里漏那里补又是一两小时，前五次都还可以，而且在电梯接下来的复用和构造中，我的设计完美地减少了代码重构量（如果不是强行要求继承的话在原方法里面改改条件就能变成ALS了）。

然而唯一美中不足的就是第六次了，但是唯一设计的最是细节完善的，也是第六次。通过这次作业我认识到，时间也是实际工程中需要考虑的一大要素。我们是要追求高质量的东西，但有时候现实条件客户可能只需要一个Just work的东西先凑合用着，之后再一次次发布新版本，发布补丁，发布迭代包（然后给市场部开发新升级服务的叫卖机会）。

我在第六次作业的前期设计上花了过多的时间，一则因为指导书存在太多疑点而且难以理解，二则第六次作业要求功能的操作较为复杂难以短时间给出较好的实现（可以尽量减少side effects的实现）。于是我一直持续关注着大班群的讨论和讨论区，一直到周一晚上才将将写了一个开头，周二晚上正式开始写又碰到了怎么保存文件系统快照的问题。讨论区里学姐说用多叉树保存文件系统结构即可，我当时觉得这是一个很高端很有效的操作，后来C语言的树就没学好的我花了一晚上学着写了很丑的树，debug到凌晨两点才开始写真正的文件系统快照对比和触发器。时间明显是来不及的。最后提交的时候碰到了奇怪的Crash一直没有弄清楚逻辑上哪里出了问题，变成了只能判定非法输入的无效作业。

后来，回顾这次作业，我觉得核心问题有两个。其一，时间控制不合理，设计应该在周日最晚周一完成，周一晚上需要开始处理核心架构的问题，周二就要结束编码，周三早上收尾。而不是担心设计漏洞和缺陷畏首畏尾；其二，文件系统架构上其实理解了指导书会发现比较文件挪动的范围很小，只是多一层目录罢了，因此直接遍历监控目录和子目录即可，比如每个监控目录做成一个数据对象，里面包含了监控的文件信息，监控目录路径，工作目录路径等基本信息。比对的时候挨个比对即可（不存在跨目录问题，可以将所有修改文件的操作放在一起合并一次后再执行）。总结以上，我觉得首先工程需要考虑很多因素的综合，是一个trade off，因此我们不能在做工程项目的时候任性只为追求设计的优雅或者自己写的爽（突然想起了《人月神话》中的外科医生团队，此处你的理性和ddl应该就是那个外科医生2333）；其次，在真实的开发过程中尽量在自己独立思考的基础上再去批判性接受他人的建议，尤其是当他人的建议会将让你引入新的领域知识的时候，需要谨慎对待它带来的潜在学习成本和时间成本。

后来的作业吸取了第六次作业的教训，早早就开始设计并开始编码才避免了产生更多问题。出租车部分最让我开心的是第一次出租车我就开创性地将所有出租车的全部移动轨迹进行了记录，精确到每一秒的运行，并在时间的校准上花费了很大的功夫（最终达到了和理论推导完全契合的精确模拟）。这让我在后来添加粉色出租车上省了很多工夫。另外一点则是我为了让调度器和出租车功能尽量逻辑上分离，我将地图和寻路作为一个GPS模块加入到了每一个出租车中（有时候模拟现实情况就是最好的设计方案，现实生活的经验方法总会在人们的使用和反馈中不断自动优化），这样，当需要更改道路的时候只要更新车辆GPS就好，而固定道路的话则将出租车GPS更新属性禁用即可——完美。寻路算法此处我推荐A\*，快速，好实现，（我是对照着网上的python源码改写的），双关键字很容易添加，逻辑清晰。

总体来看，我的设计思想可以归结为以下几条：

1. 尽量将功能模块化，然后将模块化的功能加上管理的数据封装为某个对象（这样更新功能或者更改数据都方便）最直接的应用就是我将地图和寻路算法一起封装为GPS模块。

2. 减少不必要的显式多线程（比如托盘模型完全可以用阻塞队列实现不需要自己加锁）3. 相信Java设计好的类库，尤其是线程锁方面的库（当你有某个奇怪的设计思路需要一些特殊的锁来实现，尽量寻找现成的库，他们的实现相信会比菜鸡的我们写得优秀得多）

4. 提前规划好设计思路，尽可能在设计阶段多地考虑可能碰到的问题，并在草稿纸上解决，同时注意时间

5. 对于多线程，尽量不将需要强数据相关的两个或多个类设计为需要加锁的线程。比如出租车有的同学做了100个线程，但是时间上指导书要求精确模拟，这样多线程很难实现而且调度过程中需要不停查询多个线程的运行状态，导致的结果就是很容易性能不足带不动。分析运行逻辑会发现实际上每个线程也只是按照时间粒度来更新位置（相当于100个重复的相同动作逻辑）并在接收到请求的时候才改变到一些特殊的状态。那么，我们完全可以将出租车作为数组放进调度器中，按照时间粒度运行调度器并更新出租车状态。

6. 多线程的划分有时候不是根据字面意义，而是根据功能和数据需求。“他们独立且同时运行所以他们要做成多线程”，比如100辆车就不是的。程序跑得很快，所谓同时大部分时候都是不存在的。只是模拟的粒度不一样——代码可见的模拟和CPU调度中的线性性，（除非你有100核处理器，巧妙地一个核跑一个车）只要模拟的粒度小于用户的实际需求最小时间单位。能够给用户一个***平滑***的体验，那我们就满足了“同时”的要求。我的就划分为Readin，Light，Scheduler三个线程，读取和调度之间有一个阻塞队列托盘，调度只查询灯的状态并且在每一轮时间片开始前查询全部的路灯状态并存储下来这个时刻（减少多辆出租车运行中再查询其他线程数据）。可以看出，各个线程尽可能保持隔离。

（接下来讲讲测试验证）

互测方面我的扣分秉持着代码逻辑第一的原则，测试问题尽量追溯到代码，比较典型的一次是根据对方的代码逻辑做了全部互测工作。后期的类规格和测试虽然写得着实烦躁，但对于逻辑验证推理和实现之间的关系还是有了很深的理解。自己脑补出来的部分东西和要求梳理时重新整理是完全不一样的概念。有点类似于infoGAN那样，真实的代码效果和你期望的效果就是输入。经过自编码器重构，你会发现原先复杂的耦合分布（代码）中真正重要的那几个参数。写JSF就是将复杂逻辑解耦的过程。测试则是类似于解码器的工作，将你的类规格再次合并成程序进行测试。两厢之下，你就可以发现代码中的问题，发现类规格中的问题，精简并调优逻辑。

最后一次实验课是让我印象最深的，我没曾想到可以直接从类图中直观地生成代码，甚至老师后来告诉我还可以“画”代码，也就是生成内容。由此我也在想，表达同样的一件事情，实现同样的一件事情，到底是抽象程度越高越好，还是低了好？或者某种条件下的动态平衡更为合适？在我看来，根据不同的任务需求，抽象程度需求是不一样的。抽象是为了实现方便，屏蔽一些无关紧要的细节。这里的无关紧要，计组中的汇编在OO看来就是无关紧要，c语言写一个链表实现在java中就是无关紧要，同样地，画类图我们可以快速理清类之间的关系，本身java框架性的东西很多可以自动生成，这就是无关紧要。

但是其中的逻辑如果也用画图的形式来表达的话，未免有点多此一举了，从整个理解过程来看，我们把高度抽象的设计思路表达成另一种（甚至是更高的）高度抽象（图），然后生成代码这一低维度表达。两个高维度在转换过程中的任何一点小偏差都会在低维中被放大，而且本身我们自己也已经做了整个转换的运算过程。我相信两个高维之间的脑内运算和高维到低维的映射相比，后者应该更加容易。

（理解可能不够准确还希望老师看到了能给予指正~）

总的来说我觉得课程还是很好的，尤其是工程能力的培养。就是不同作业以及整体选课的难度梯度和时间上需要仔细和其他课程协同一下，毕竟这是一个课程体系，有的作业比如后几次JSF的互测压力过大，但是本身内容应该是作为一个了解和入门使用的级别来学习（形式化验证老师您最后一节课也说“让你们体验一下，其实真正到了公司里也就这些东西，你们就熟悉的多了”），过多的时间分配使得OS和离散没办法得到充分的时间，更何谈科研和自身的课外学习（冯如杯什么的完全没时间继续深入了有点可惜）。

能同时做好好几件事，睡觉少了还照学不误的大佬还是很少的。因此强烈建议难度分级，不同的课程和科研任务之间完全可以凭个人兴趣组合选课难度。G点这种东西不应该成为限制大家发展兴趣的束缚，尤其是在大学（应该是精英教育），一个作为发展职业兴趣方向，培养科研志趣，以大部分课程的通识教育为主要目标，自由选择特定领域深入的地方。（印象很深的是国外大学生GPA上3.8，3.9似乎是司空见惯的事情，考核标准低了，但是也没见得他们培养出来的精英会比我们少）

而我们现在做的似乎更像是九年义务教育的延伸。

一点牢骚，一点总结，一点感悟，如果说这门课程除了知识还能让大家明白点什么，坚持点什么，我希望是——

“独立之精神，自由之思想。”