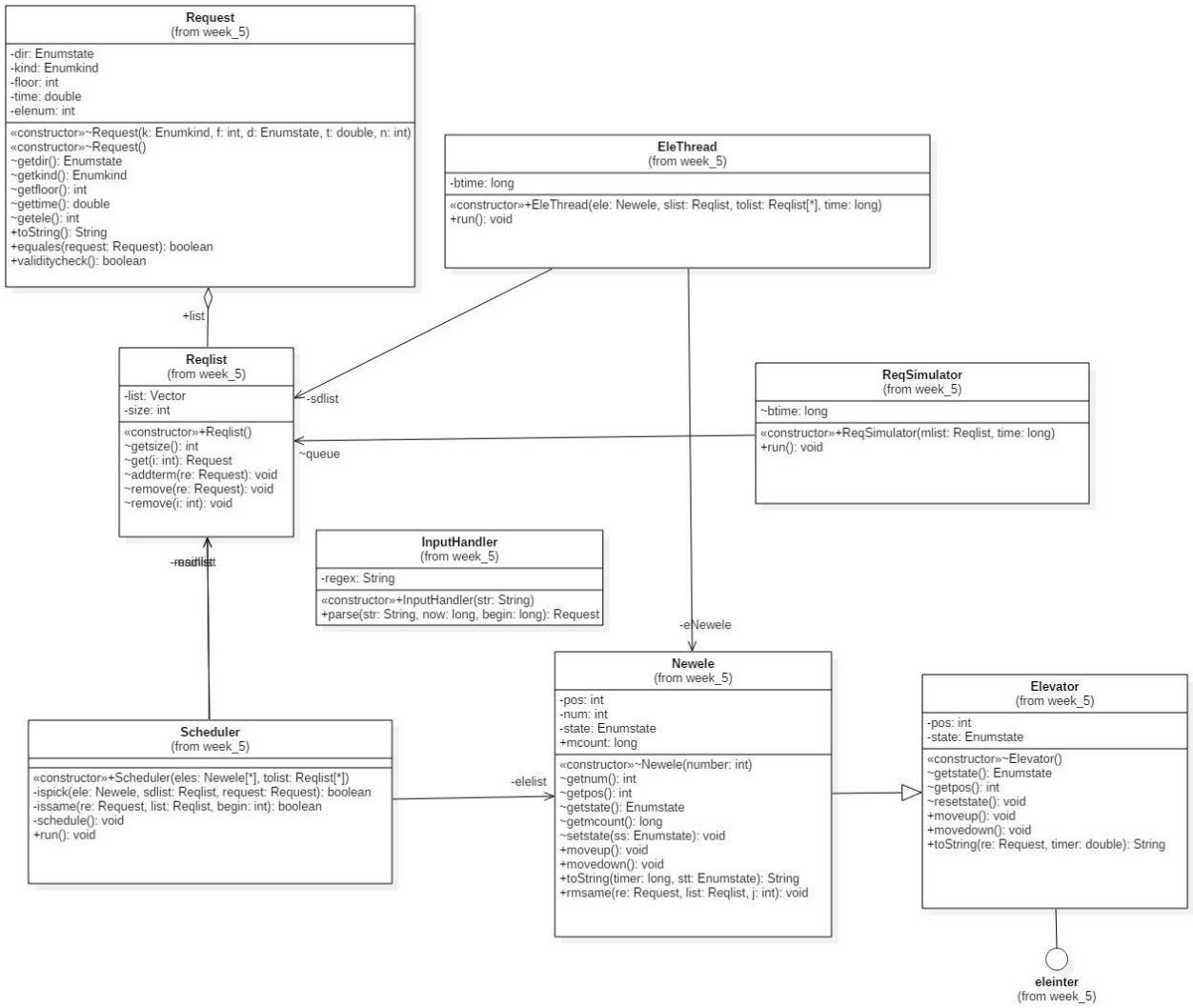
作业总结

一、 第五次作业

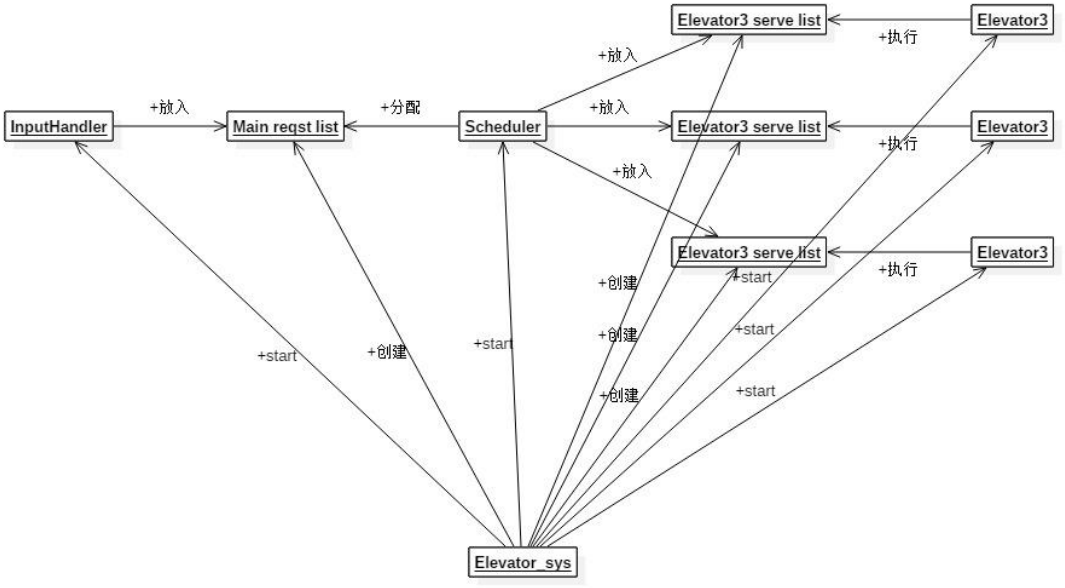
1. 度量类的属性个数、方法个数、每个方法规模、每个方法的控制分支数目、类总代码规模

Class	属性数量	方法数量	方法规模	方法分支数	类规模
Elevator	2	7	4-15	0-4	61
NewEle	4	10	3-12	0-2	71
InputHandler	1	2	2,31	0,2	40
Request	5	10	3-12	0-2	85
Reqlist	2	6	3-4	0	36
RequestSimulator	2	2	4,34	3	58
Scheduler	3	5	10-105	0-8	172
EleThread	3	2	5,55	0,7	72

2. 类图如下：



3. 线程协作示意图：



4. 设计优缺点
   1) 优点：采用了生产者消费者模式，结构清晰、合理，共享数据清楚。
   2) 缺点：没有使用线程阻塞的方法(wait 和 notify)，而是让线程长期处于忙等待状态，导致 CPU 性能的浪费。调度线程将电梯线程当做对象引用，不如将电梯线程的状态发布到 status board 的设计方法清楚。
5. Bug 分析：

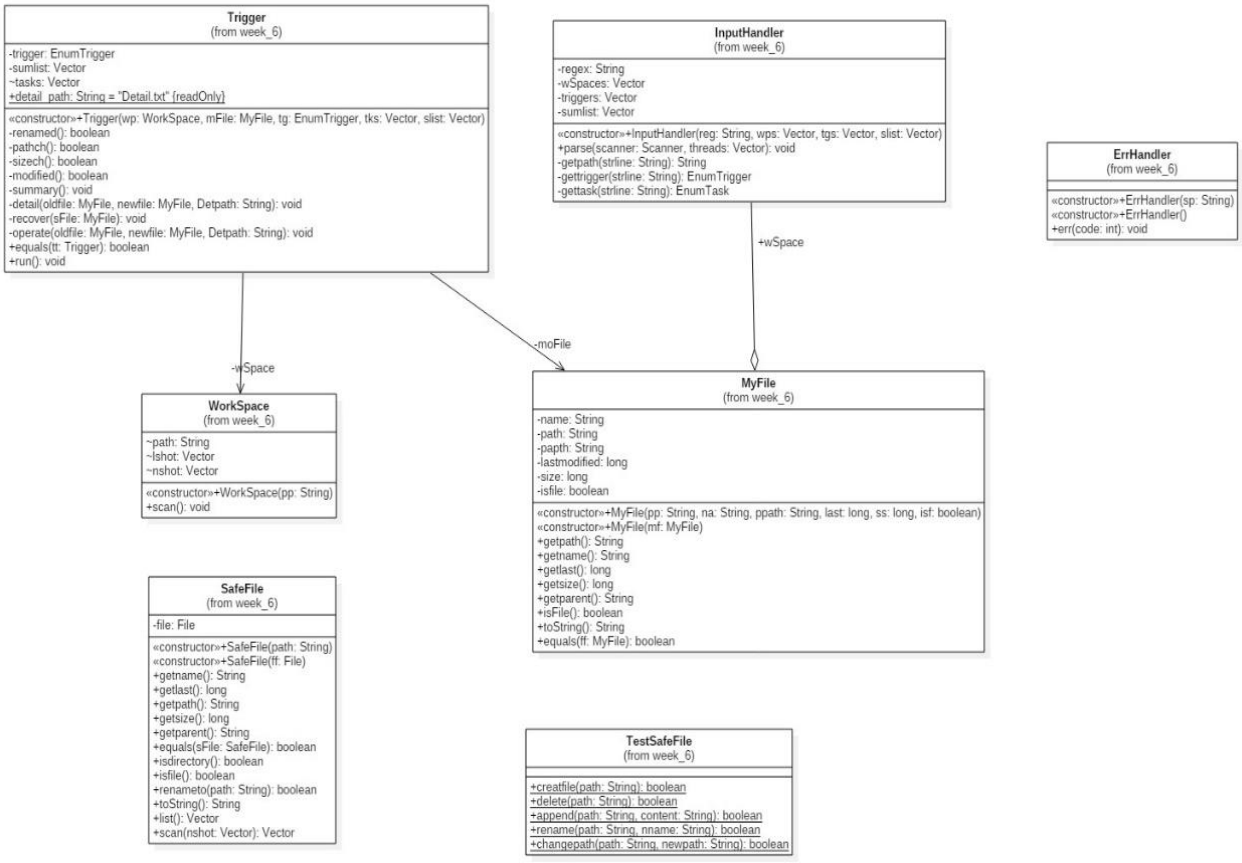
   本次作业暂未发现 bug。

二、 第六次作业

1. 度量类的属性个数、方法个数、每个方法规模、每个方法的控制分支数目、类总代码规模

Class	属性数量	方法数量	方法规模	方法分支数	类规模
MyFile	6	10	2-6	0-2	65
SafeFile	1	14	2-20	0-3	91
InputHandler	4	5	4-90	0-10	204
Trigger	5	11	10-45	0-4	241
TestSafeFile	0	5	15-30	0-3	79
WorkSpace	3	2	10,25	0,1	43
Main	0	1	50	1	74
Summary	3	4	3-6	0	35

2. 类图如下：



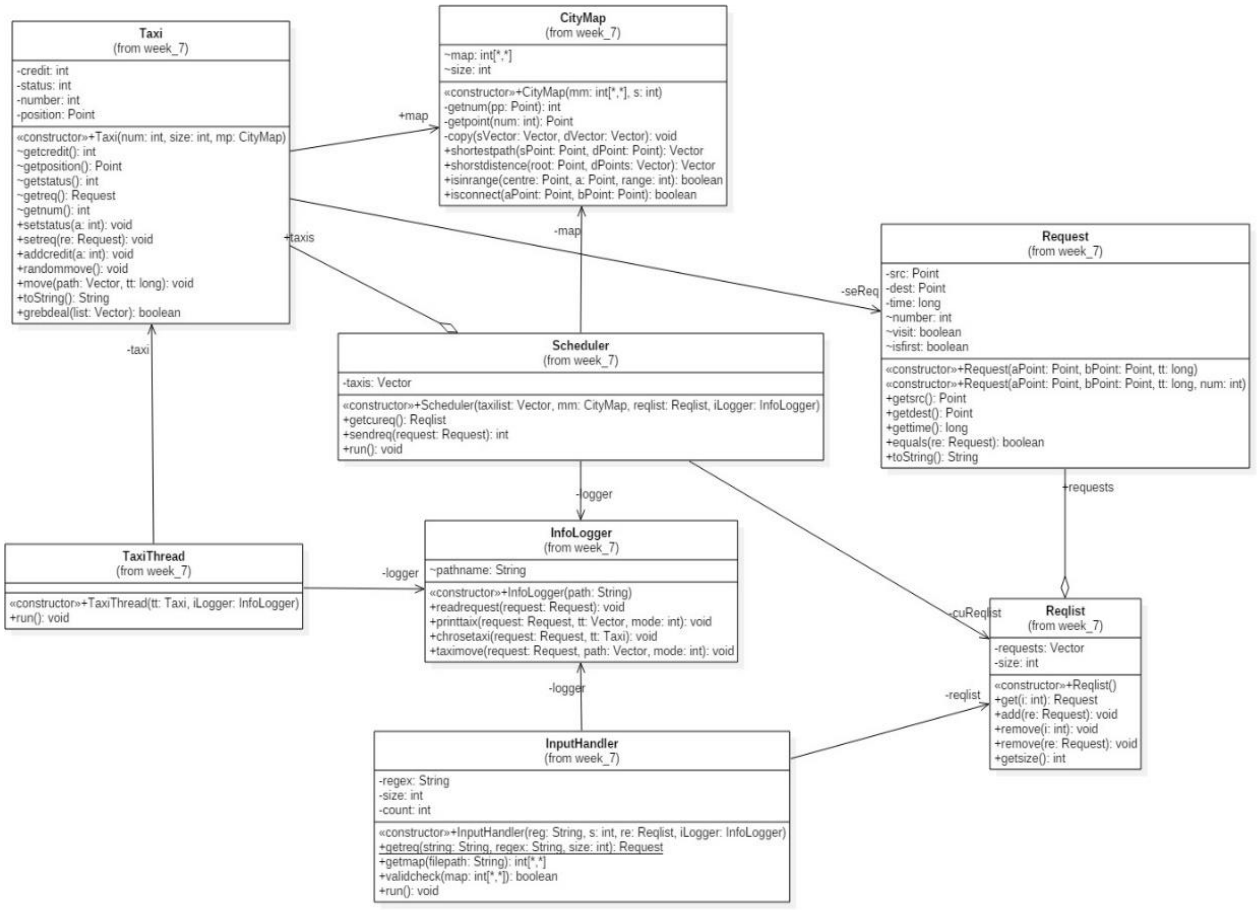
3. 设计优缺点：
   1) 优点：线程数目较少，结构清楚；
   2) 缺点：对快照的数据结构没有设计好，直接采用队列来存储导致对比出现困难，一些触发器不能触发。
4. Bug 分析：
   1) 一些触发器不能触发。快照的结构没有设计好，对比时一些特殊情况没有考虑，导致出现这个问题。应当构造一个文件树，而不能简单的采用队列来存储。

三、 第七次作业

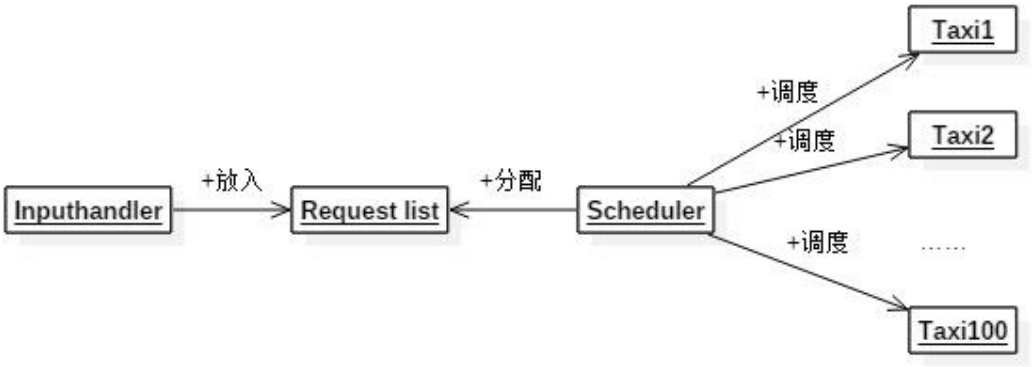
1. 度量类的属性个数、方法个数、每个方法规模、每个方法的控制分支数目、类总代码规模

Class	属性数量	方法数量	方法规模	方法分支数	类规模
Taxi	6	13	3-25	0-4	151
CityMap	2	8	4-40	0-5	145
InputHandler	5	5	8-50	0-5	164
Request	6	7	3-10	0	54
Reqlist	2	6	4-5	0	34
InfoLogger	2	5	3-40	0-3	135
Scheduler	4	4	10-70	0-5	130
TaxiThread	2	2	3-50	0-2	65
GUIThread	4	2	10-30	2	43
Taxi_sys	0	1	55	2	66

2. 类图如下：



3. 线程协作示意图如下：



4. 设计优缺点：
   1) 优点：采用共享对象的方法完成线程之间的信息交流，结构清楚。
   2) 缺点：GUI 与出租车状态的切合度不够，二者之间存在误差。为测试留下的测试接口存在与 console 同时输入的同步问题。
5. Bug 分析：
   暂未发现 bug。

四、 测试方法

首先应当反思的是，虽然每次作业都为测试者准备了测试方法，但这些方法自己都没有使用过，也就是说，我只使用了控制台输入的方法来测试自己的程序，其中既有时间因素，也有个人的惰性。这也是我第六次作业出现较多 bug 的原因。

在测试阶段，从第七次作业开始我采用了编写测试线程和控制台输入结合的方法来测试（第六次作业拿到的是无效作业，所以没有进行测试工作）。鉴于本次作业的难度不大，主要在一个最短路径算法，如果这个功能已经实现了，就很难通过手动输入的方式找到 bug。通过编写测试线程的方法，确实能够发现一些细微之处的问题。线程之间的同步问题在此时很容易被暴露出来。

五、 心得体会

在设计合理的情况下，多线程并不可怕。

线程之间的通信非常重要，必须为共享信息做好同步控制。

应当在用阻塞和唤醒的方法来避免线程的忙等待。