第 13 次作业 readme

一、 第三次作业 readme

a) 电梯调度策略和程序功能说明：

调度策略：

当请求队列不为空，选取一个请求作为主请求。判断该主请求的方向（UP，DOWN， STILL），根据不同情况判断捎带请求和同质请求。

功能说明：
实现可以捎带的电梯

b)程序运行所需环境和运行指令规范

运行所需环境： eclipse

运行指令规范：

调度器从 console 持续读取输入，直到输入为 RUN 时或有效指令（不报ERROR 的指令）到 100 条时。对每个有效请求构造请求对象，存入请求队列。
当请求队列不为空，选取一个请求作为主请求。电梯每到一层判断是否有该层的捎带请求。执行该请求后，需要从队列中 remove 当前请求，并 remove 该
请求的同质请求，直到主请求执行完毕，根据条件选出下一主请求。直到队列为
空，电梯调度完毕。

c)程序输入说明

标准输入格式：

电梯类请求（ER， n， T）

楼层类请求（FR， n， UP/DOWN， T）

输入限制：

其中， n 为楼层，可有前导 0，最多支持 10 个前导 0，可有前导+。

T 为时刻，非负整数，最多支持 10 个前导 0，可有前导+。

遇见输入错误时的相应信息：

1.对于时刻数超出最大边界的，报出 ERROR， #Time out

2.对于时间乱序的（当前请求时刻小于之前有效请求或者第一条请求时刻不为 0），报出 ERROR， #Time disorder

3.对于除 1,2 之外的无效请求，报出 ERROR， #Matching Error

d)程序输出规范&可预见运行错误响应信息

输出规范：

无效请求： INVALID[request]， request 为删掉空格的无效请求

同质请求：#SAME[request]，request 为标准输出，电梯请求为：(ER,m,T)，楼层请求为(FR,m,DOWN/UP,T)，其中中括号里无空格， 无小括号， 且数字前

无前导 0 和前导+,数字都为整数型。

有效请求： [request]/(n,UP/DOWN/STILL,T)， request 为与同质请求相同的标

准输出， 中括号里无小括号，无空格， n 为 1-10 间整数，无前导 0 和前导+， T

无前导 0 和前导+，保留到小数点后一位。

可预见运行错误响应信息：

1.合法指令正常输出，楼层数和时刻数最多支持 10 个前导 0，支持前导+。

2.规定只处理前 100 条指令，当达到 100 还没有 RUN 时，程序直接处理前一百

条请求并输出结果。

3.对于时刻数超出最大边界的，报出 INVALID[request]

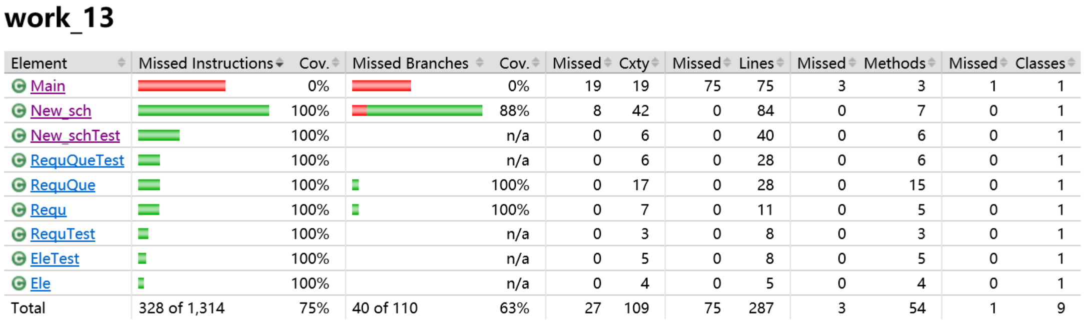
4.对于时间乱序的（当前请求时刻小于之前有效请求或者第一条请求时刻不为 0），

报出 INVALID[request]

5.对于除 3,4 之外的无效请求， 直接报出 INVALID[request]

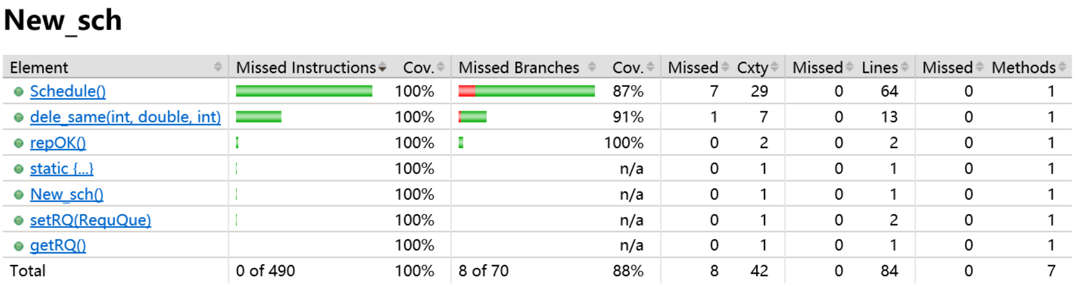
6.对于直接输入 RUN 的程序，没有输出，程序结束。

二、 运行覆盖率报告



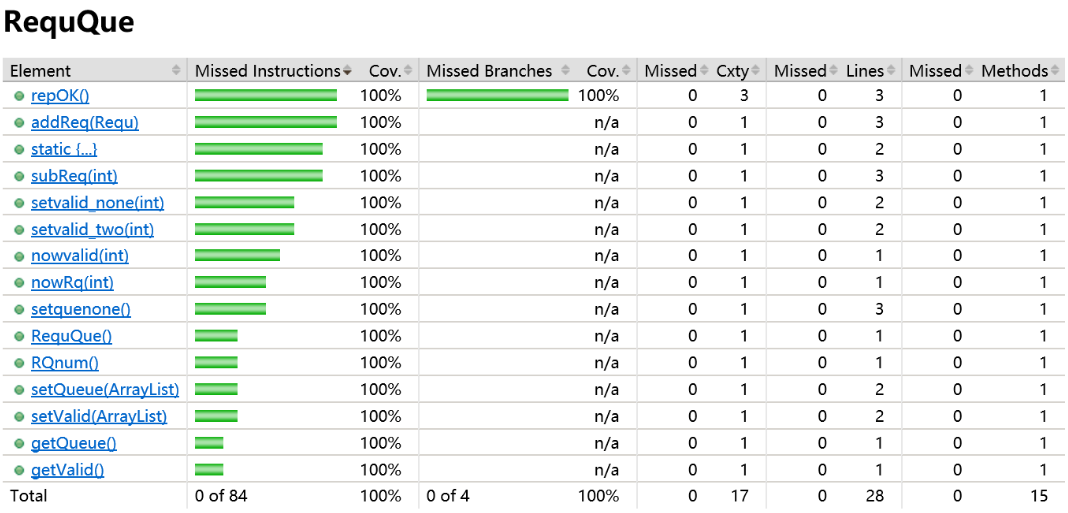
work_13

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
Main		0%		0%	19	19	75	75	3	3	1	1
New_sch		100%		88%	8	42	0	84	0	7	0	1
New_schTest		100%		n/a	0	6	0	40	0	6	0	1
RequQueTest		100%		n/a	0	6	0	28	0	6	0	1
RequQue		100%		100%	0	17	0	28	0	15	0	1
Requ		100%		100%	0	7	0	11	0	5	0	1
RequTest		100%		n/a	0	3	0	8	0	3	0	1
EleTest		100%		n/a	0	5	0	8	0	5	0	1
Ele		100%		n/a	0	4	0	5	0	4	0	1
Total	328 of 1,314	75%	40 of 110	63%	27	109	75	287	3	54	1	9



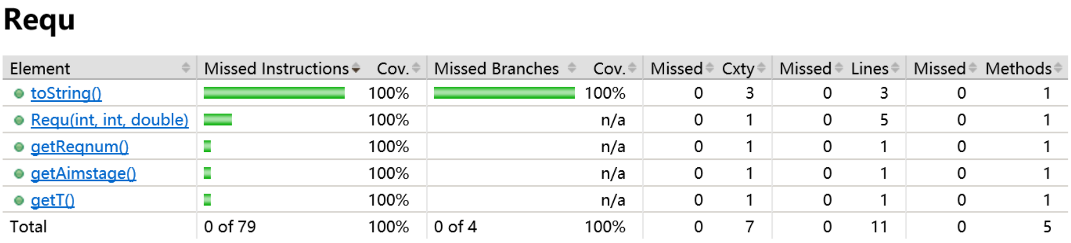
New_sch

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods
Schedule()		100%		87%	7	29	0	64	0	1
dele_same(int, double, int)		100%		91%	1	7	0	13	0	1
repOK()		100%		100%	0	2	0	2	0	1
static {...}		100%		n/a	0	1	0	1	0	1
New_sch()		100%		n/a	0	1	0	1	0	1
setRQ(RequQue)		100%		n/a	0	1	0	2	0	1
getRQ()		100%		n/a	0	1	0	1	0	1
Total	0 of 490	100%	8 of 70	88%	8	42	0	84	0	7



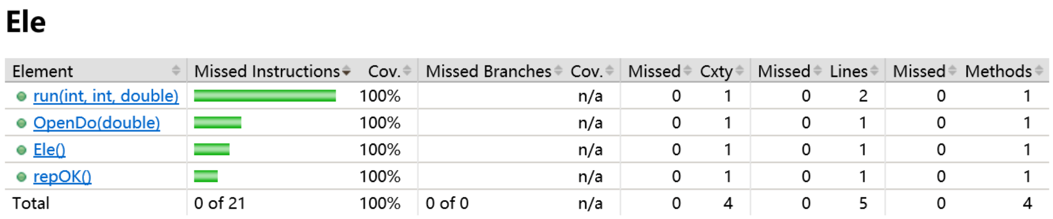
RequQue

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods
repOK()		100%		100%	0	3	0	3	0	1
addReq(Requ)		100%		n/a	0	1	0	3	0	1
static {...}		100%		n/a	0	1	0	2	0	1
subReq(int)		100%		n/a	0	1	0	3	0	1
setvalid_none(int)		100%		n/a	0	1	0	2	0	1
setvalid_two(int)		100%		n/a	0	1	0	2	0	1
nowvalid(int)		100%		n/a	0	1	0	1	0	1
nowRq(int)		100%		n/a	0	1	0	1	0	1
setquenone()		100%		n/a	0	1	0	3	0	1
RequQue()		100%		n/a	0	1	0	1	0	1
RQnum()		100%		n/a	0	1	0	1	0	1
setQueue(ArrayList)		100%		n/a	0	1	0	2	0	1
setValid(ArrayList)		100%		n/a	0	1	0	2	0	1
getQueue()		100%		n/a	0	1	0	1	0	1
getValid()		100%		n/a	0	1	0	1	0	1
Total	0 of 84	100%	0 of 4	100%	0	17	0	28	0	15



Requ

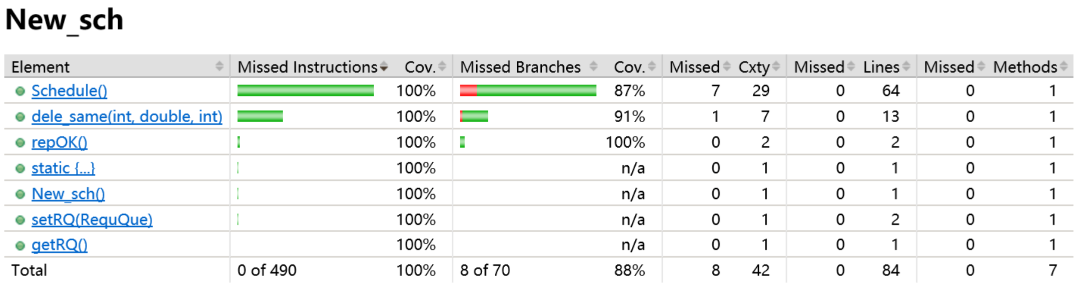
Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods
toString()		100%		100%	0	3	0	3	0	1
Requ(int, int, double)		100%		n/a	0	1	0	5	0	1
getReqnum()		100%		n/a	0	1	0	1	0	1
getAimstage()		100%		n/a	0	1	0	1	0	1
getT()		100%		n/a	0	1	0	1	0	1
Total	0 of 79	100%	0 of 4	100%	0	7	0	11	0	5



Ele

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods
run(int, int, double)	🟩	100%		n/a	0	1	0	2	0	1
OpenDo(double)	🟩	100%		n/a	0	1	0	1	0	1
Ele()	🟩	100%		n/a	0	1	0	1	0	1
repOK()	🟩	100%		n/a	0	1	0	1	0	1
Total	0 of 21	100%	0 of 0	n/a	0	4	0	5	0	4

三、 测试结果



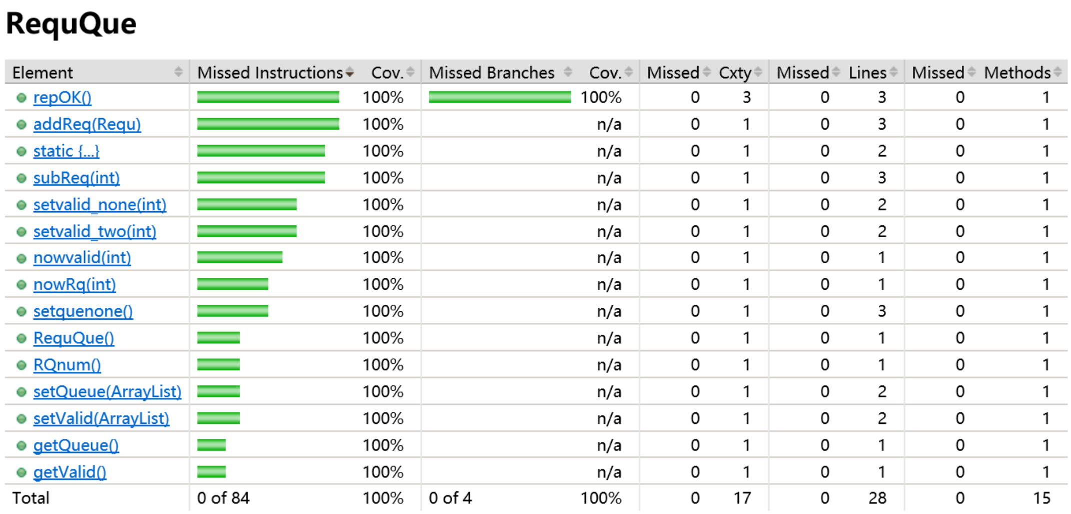
New_sch

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods
Schedule()	🟩	100%	🟥🟩	87%	7	29	0	64	0	1
dele_same(int, double, int)	🟩	100%	🟥🟩	91%	1	7	0	13	0	1
repOK()	🟩	100%	🟩	100%	0	2	0	2	0	1
static {...}	🟩	100%		n/a	0	1	0	1	0	1
New_sch()	🟩	100%		n/a	0	1	0	1	0	1
setRQ(RequQue)	🟩	100%		n/a	0	1	0	2	0	1
getRQ()		100%		n/a	0	1	0	1	0	1
Total	0 of 490	100%	8 of 70	88%	8	42	0	84	0	7

方法名	testSchedule
测试用例数目	1
分支覆盖率	87%
语句覆盖率	100%
复现 bug 数	0
新发现 bug 数	0

方法名	testdele_same
测试用例数目	1
分支覆盖率	91%
语句覆盖率	100%
复现 bug 数	0
新发现 bug 数	0

方法名	testrepOK
测试用例数目	3
分支覆盖率	100%
语句覆盖率	100%
复现 bug 数	0
新发现 bug 数	0



RequQue

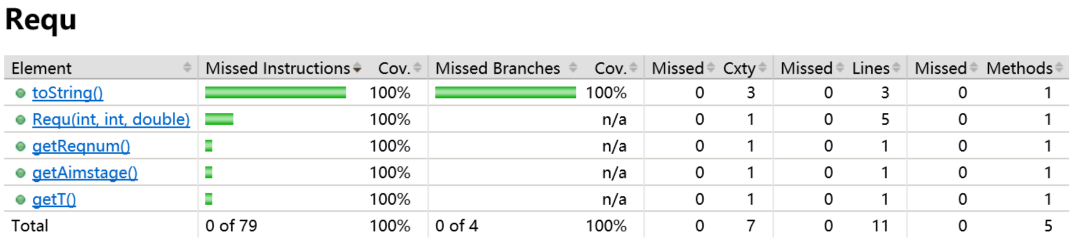
Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods
repOK()	🟩	100%	🟩	100%	0	3	0	3	0	1
addReq(Requ)	🟩	100%		n/a	0	1	0	3	0	1
static {...}	🟩	100%		n/a	0	1	0	2	0	1
subReq(int)	🟩	100%		n/a	0	1	0	3	0	1
setvalid_none(int)	🟩	100%		n/a	0	1	0	2	0	1
setvalid_two(int)	🟩	100%		n/a	0	1	0	2	0	1
nowvalid(int)	🟩	100%		n/a	0	1	0	1	0	1
nowRq(int)	🟩	100%		n/a	0	1	0	1	0	1
setquenone()	🟩	100%		n/a	0	1	0	3	0	1
RequQue()	🟩	100%		n/a	0	1	0	1	0	1
RQnum()	🟩	100%		n/a	0	1	0	1	0	1
setQueue(ArrayList)	🟩	100%		n/a	0	1	0	2	0	1
setValid(ArrayList)	🟩	100%		n/a	0	1	0	2	0	1
getQueue()	🟩	100%		n/a	0	1	0	1	0	1
getValid()	🟩	100%		n/a	0	1	0	1	0	1
Total	0 of 84	100%	0 of 4	100%	0	17	0	28	0	15

方法名	testrepOK
测试用例数目	3
分支覆盖率	100%
语句覆盖率	100%
复现 bug 数	0
新发现 bug 数	0

方法名	testAddReq
测试用例数目	1

分支覆盖率	100%
语句覆盖率	100%
复现 bug 数	0
新发现 bug 数	0

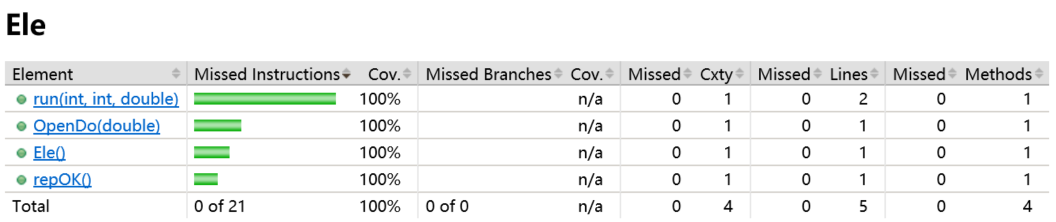
方法名	testSubReq
测试用例数目	1
分支覆盖率	100%
语句覆盖率	100%
复现 bug 数	0
新发现 bug 数	0



Requ

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods
● toString()	🟩	100%	🟩	100%	0	3	0	3	0	1
● Requ(int, int, double)	🟩	100%		n/a	0	1	0	5	0	1
● getReqnum()	🟩	100%		n/a	0	1	0	1	0	1
● getAimstage()	🟩	100%		n/a	0	1	0	1	0	1
● getT()	🟩	100%		n/a	0	1	0	1	0	1
Total	0 of 79	100%	0 of 4	100%	0	7	0	11	0	5

方法名	testToString
测试用例数目	3
分支覆盖率	100%
语句覆盖率	100%
复现 bug 数	0
新发现 bug 数	0



Ele

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods
run(int, int, double)	🟩	100%		n/a	0	1	0	2	0	1
OpenDo(double)	🟩	100%		n/a	0	1	0	1	0	1
Ele()	🟩	100%		n/a	0	1	0	1	0	1
repOK()	🟩	100%		n/a	0	1	0	1	0	1
Total	0 of 21	100%	0 of 0	n/a	0	4	0	5	0	4

方法名	testrepOK
测试用例数目	1
分支覆盖率	100%
语句覆盖率	100%
复现 bug 数	0
新发现 bug 数	0

方法名	testRun
测试用例数目	1
分支覆盖率	100%
语句覆盖率	100%
复现 bug 数	0
新发现 bug 数	0

方法名	testOpenDo
测试用例数目	1
分支覆盖率	100%
语句覆盖率	100%
复现 bug 数	0

新发现 bug 数	0