

# 代码重构

本次作业在第三次作业的基础上进行了重构，在获得助教允许的前提下取消了对第二次调度方法的继承机制。

# 输入与输出

请输入与输出的要求与第三次作业相同，请参考上传的第三次作业Readme

# bug报告

程序在第三次作业测试以及本次测试中均为发现bug。

# 测试用例及测试结果概述

为方便对同一方法的多次测试，避免多次重复构建测试环境以及准备相关测试用例，本程序对于同一方法的多个测试用例都写在了一个测试方法中，按照这个测试方法调用被测方法的次数来统计测试用例的个数。

由于repOK方法本身不变式的特殊性，为覆盖repOK方法的所有分支，在调用其他方法构造测试用例时没有遵循对应方法的REQUIRES前置条件。在测试其他普通方法时，根据Guideline文档的规定，程序仅保证调用者在遵循REQUIRES条件和不变式的情况下所得到的方法结果的正确性，在不遵循REQUIRES条件以及不变式的情况下所得到的方法结果不一定正确。故对于普通方法的测试仅考虑满足前置条件和不变式的情况。

除get，set类方法，main方法外本次测试的语句覆盖率和分支覆盖率均达到了100%，详细的测试结果在report文件夹中。

## Main类

所测试的方法名	测试用例数目	语句覆盖率	分支覆盖率	所复现的bug数	新发现的bug数
无					

该类中只有一个main方法，且main方法不用测试。

## ALS\_Scheduler类

所测试的方法名	测试用例数目	语句覆盖率	分支覆盖率	所复现的bug数	新发现的bug数
repOK	7	100%	100%	0	0
ALS_Scheduler构造方法	4	100%	n/a	0	0
updateStatus	7	100%	100%	0	0
runStill	4	100%	n/a	0	0
runUpOrDown	2	100%	100%	0	0
checkSameRequest	2	100%	100%	0	0
checkPiggybacking	3	100%	100%	0	0
changeMainRequest	7	100%	100%	0	0

该类中checkSameRequest等方法虽然有较多分支，但因为错在循环语句，故可以通过在请求队列中加入多个请求，然后用少量的调用次数覆盖多数分支。

## *Elevator*类

所测试的方法名	测试用例数目	语句覆盖率	分支覆盖率	所复现的bug数	新发现的bug数
repOK	9	100%	100%	0	0
toString	21	100%	n/a	0	0

该类中大多数为get和set类方法，所以并未给这些方法创建单独的测试用例。

## *Floor*类

所测试的方法名	测试用例数目	语句覆盖率	分支覆盖率	所复现的bug数	新发现的bug数
无					

该类为空，没有任何方法。

## *Request*类

所测试的方法名	测试用例数目	语句覆盖率	分支覆盖率	所复现的bug数	新发现的bug数
Request构造方法	7	100%	n/a	0	0
checkFormat	33	100%	100%	0	0
toString	4	100%	100%	0	0

该类中大多数为get和set类方法，所以并未给这些方法创建单独的测试用例。

### RequestQueue类

所测试的方法名	测试用例数目	语句覆盖率	分支覆盖率	所复现的bug数	新发现的bug数
repOK	6	100%	100%	0	0
enQueue	1001(2+999)	100%	100%	0	0
deQueue	3	100%	100%	0	0
isEmpty	2	100%	100%	0	0
isFull	2	100%	100%	0	0

该类中多数为get和set类方法，所以并未给这些方法创建单独的测试用例。

该类中对enQueue进行测试时为构造队列为满的条件，循环执行了999次enQueue方法，如果认为这999次调用不属于独立的测试用例，则也可认为测试用例数目为2。

### 其他说明

根据助教的回答，对于在一个方法内调用其他方法更改属性，可以不用写在本方法的jsf中的EFFECTS和MODIFIES中。