

JAVA 说明文档

一、程序功能说明

该程序为一个实现了接客、修路、红绿灯、可追踪出租车的出租车模拟程序。

二、程序运行环境

JAVA 版本: 1.8.0_111

Eclipse 版本: Mars.1 Release (4.5.1)

三、输入说明

控制台部分输入请参见指导书。

此次作业新增了 ER 请求[ER,(i,j),dir]。其中 ER 为标识符, i 和 j 为节点坐标, 坐标的编号为 1~80(和指导书相同, 和 GUI 不同); dir 为边位于该节点的位置, 只能是 RIGHT 或者 DOWN, 分别代表节点右面的边和下面的边。

eg: [ER,(40,40),RIGHT]

和 CR 请求一样, 每 100ms 内发出的请求会被判定为同时发出。ER 请求生效时会将该边的存在性进行反转——如果该边存在, 那么会被变更为不存在; 如果不存在, 会被变更为存在。

四、输出说明

输出文件在“data.txt”中, 标题为请求, “Taxis' information”为所有请求输入时刻附近的出租车, “All Grabbing Taxis”为所有抢单的出租车, “Taxi selected”为被钦定的出租车, “Taking Passenger”为接乘客路径, “Serving”为服务路径。

如果在修改路的时候有出租车在路上, 请求虽然合法, 但不会生效, 并会提示“Taxis on the edge!”。

五、设计说明

出租车的编号为 0~99, 坐标的编号为 1~80 (和 GUI 不同)。

车辆的状态分为四种: STOPPING, SERVING, WAITING, TO_PASSENGER (参考 enum taxiState), 分别对应着停止 (0), 服务 (1), 等待服务 (2), 接单 (3)。

程序没有执行完毕的时候打开 data.txt 有几率由于文件占用冲突导致信息无法写入, 请保证至少所有出租车都处于 WAITING 状态再确认信息。

main 函数在 Runner 类, 其中有个 SHOW_CR_GUI 参数, 如调成 true, 会显示请求窗口, 如果阁下觉得弹窗神烦可以调成 false……

如果接到了合法的 ER 请求, 那么所有有明确目的地的出租车会重新寻路。

非交叉路口节点无法安装红绿灯, 因此如果某交叉路口因为删边变成了非交叉路口, 那么这个路口的红绿灯会被删掉, 同时, 即便之后再在这个路口把删掉的边加上, 失去的红绿灯也不会再回来了。(严格按照指导书对于“交叉路口”的要求, 不考虑新出租车行驶关闭边的问题)

对于流量里时间窗的说明有些模糊, 我的理解是“经过”是指出租车在边上, 当出租车离开边之后, 其对应的流量会在 200ms 后被减去。

无论何种状态, 如果出租车在等灯的时候要走的边被拆了, 那么它回来等灯结束之后重新规划路线。

处于 WAITING 状态的出租车等灯的时间不计算在 20s 内。

关闭的道路虽然流量为 0, 但通过关闭并在 200ms 内打开道路并不能达到流量清零的效果（我认为这样更符合实际）。

我对关闭的道路的定义是“存在过但是现在不存在的道路”，因为新出租车不是飞机，不可能凌驾于地图之上 orz

新出租车的乘客服务情况就是输出到 data.txt 的信息（是为了更全面输出信息，才不是偷懒呢），指导书说的那些信息我都包括了，求饶过~

如果在不满足 hasNext 的情况下调用 next，会返回 null，previous 同理。

六、规格说明

“[]”中间填的是中间变量；

MODIFIES 后面注明的对象是在方法中可能被更改的，不是一定被更改；

描述中对于一些方法比较抽象，如 contains 修饰的不一定是容器，也可能修饰的是有某个容器属性的对象，或者是个数组。

第十次作业新增了类的 Overview、不变式和 repOK()，其中 InfoHandler、Runner 和 TestThread 三个类没有不变式和 repOK()，原因在于前两个类无需构造实例，而 TestThread 由测试者编写，故不存在不变式。同时对于一些无属性的枚举类，不变式也没有被注明。

对于接口，只提供类的 Overview，其方法的规格和具体实现有关，这里不做注明。

七、测试说明

特意为阁下在 TestThread 中准备了五种测试方法：

putCRrequest: 发出一条 CR 请求；

showTaxiInfo: 输出指定编号的出租车信息；

showStateTaxis: 输出指定状态的所有出租车编号；

getStateTaxis: 获得一个包含指定状态的所有出租车的 ArrayList；

showSysMill: 输出当前系统时间（System.currentTimeMillis()）。

以上方法已做到线程安全，敬请阁下在 run 方法中编写测试代码。

taxiInit 方法在 Runner 类中，没太明白指导书的意思，如果要重新规定顺序的话，请把我的代码注释掉吧~

调用 TraceableTaxi 的 twoWayIterator 方法获取迭代器。

八、BUG 修复

暂无 BUG。