说明文档Lab1

1. 解决思路
2. 程序运行大致顺序:
3. 启动后立即从控制台读入棋盘格式,考虑使用readBoard方法

Board.*readBoard*();

(2)读入后等待用户输入命令,分为print,at x y,exit三种,还有对非法输入的处理.

考虑使用选择结构识别执行不同命令与非法输入.

1. 对各个命令的执行

(1)程序要求在exit命令退出程序之前,可以不断读取用户命令并执行.故考虑以条件为true的while循环执行各种命令的选择,执行exit命令时,break跳出循环,结束程序.

(2)执行命令前需要识别是什么类型的命令,引入字符串command读入每次输入的命令,并做识别.print和exit命令用String.equals()识别,at x y则用String.startsWith()识别.如果command不能识别为以上三种情况,则视为非法输入,输出invalid input.

String command = scanner.nextLine();

**if**(command.equals("print"))

**else** **if**(command.startsWith("at"))

**else** **if**(command.equals("exit"))

**else**

(3)对print命令的执行:其任务为打印棋盘,调用printBoard方法打印出readBoard方法中一开始从控制台读入的棋盘.

Board.*printBoard*();

(4)对at x y命令的执行: 首先从用户的输入中找到指定棋子的坐标.用户的输入形式为at x y,用以字符串数组input读入commad中略去空格的部分,再将坐标数字部分赋值给int变量.使用getCoordinates方法得到本身及左右上下坐标及值,并添加到ArrayList中,并输出.

String[] input = command.split(" ");

**int** x = Integer.*parseInt*(input[1]);

**int** y = Integer.*parseInt*(input[2]);

List<Coordinate>adjacentCoordinates =

Board.*getCoordinates*(x,y);

System.***out***.println(adjacentCoordinates);

(5)对exit命令执行:break跳出循环,结束程序

(6)对非法输入:输出invalid input

1. 各种方法
2. readBoard:定义数组board.按行读取控制台,注意略去”,”.再按列读入每行上的棋子.为了顺序读取采取for循环.
3. printBoard:for循环打印出已读取的棋盘
4. getCoordinates:以列表显示本身及左右上下棋子的坐标及类型,选择if语句检查是否存在左右上下的棋子,若存在则添加到列表当中.
5. Coordinates:this调用成员变量x,y,item;返回字符串类型以符合列表中每个元素的格式.
6. 说明
7. 各类存放的方法:Body类中main方法;

Board类中readBoard,printBoard,getCoordinates方法;

Coordinate类中Coordinate.

1. at x y命令中先按行再按列,坐标棋子后输出的相邻棋子顺序为左,右,上,下.
2. 命令的执行:执行一种类型命令,输出一次.特别地at x y命令可以一次给出多个目标棋子并执行(因为at x y命令是一种类型的命令).

For example:

Input(之前已输入过棋盘,再用复制将一下文本一次性复制到控制台上,复制后按一次Enter键,有时也不需要按Enter键):

at 2 2

at 0 3

at 0 0

at 4 2

at 0 4

output: (复制后按下Enter键输出,有时复制后自动输出)

[(2,2,E), (2,1,D), (2,3,A), (1,2,D), (3,2,A)]

[(0,3,D), (0,2,C), (0,4,E), (1,3,E)]

[(0,0,A), (0,1,B), (1,0,B)]

[(4,2,B), (4,1,A), (4,3,C), (3,2,A)]

[(0,4,E), (0,3,D), (1,4,A)]

(注意:有时一次性执行多个at x y命令时,第一次按下Enter键后最后一个目标棋子及相邻棋子的列表不一定会输出,此时需要再按下一次Enter键)

即

Input(之前已输入过棋盘,再用复制将一下文本一次性复制到控制台上,复制后按一次Enter键):

at 2 2

at 0 3

at 0 0

at 4 2

at 0 4

output:

(第一次按下Enter键后显示:)

[(2,2,E), (2,1,D), (2,3,A), (1,2,D), (3,2,A)]

[(0,3,D), (0,2,C), (0,4,E), (1,3,E)]

[(0,0,A), (0,1,B), (1,0,B)]

[(4,2,B), (4,1,A), (4,3,C), (3,2,A)]

(第二次按下Enter键后显示:)

[(0,4,E), (0,3,D), (1,4,A)]