设计思路：

1. 打印开场白和操作说明，打印第一关地图开始第一关。
2. 获取用户输入的指令，并根据指令执行对应的方法。
3. 输出指令的执行结果。
4. 判断其结果是否是赢局、死局1或者死局2。若是，打印结果。
5. 转2。

程序结构：

Main Class: BoxMover

定义5个static变量

定义Map对象数组，不作实例化避免浪费。在每一步需要新建地图时实例化

main()

while(true){

methods.do\_order(methods.input())获取指令并执行

methods.showResult();输出结果

methods.judge();判断

}

Map Class: extends BoxMover

readMap()从本地文件中读取地图

drawMap()打印地图

drawMapElement(int element)根据数字到图形的映射关系打印地图元素

clearMap(int width, int height)清空地图数据

findPos()找到Mover在地图中的位置

getter：

getMap():int[][]获取地图

getMapElement(int i, int j):int获取地图元素

getplayerX():int获取人的横坐标，需要先执行findPos()

getplayerY():int获取人的横坐标，需要先执行findPos()

setter:

setMap(int map[][])修改地图

setMapElement(int i, int j ,int element)修改地图元素

Methods Class: extends BoxMover

Comanding Methods:

do\_order(String)组织调用指令判断、指令执行函数

showResult()输出一个指令的结果

judge()判断赢局、死局1和死局2

Function Performing Methods：

exit(), share(), illegal\_input(), restart()退出、分享、非法输入、重新开始方法

up(), left(), right(), down()上下左右

leave(int, int):int判断当前Mover脚下的是空地还是箱子目的地

first\_words(),help()开场白和帮助

isNumber(String):boolean, 判断输入的是否是1-9

changeLevel(String)换关

isRetreat(String):boolean,判断输入的是否是悔棋指令

retreat(String)悔棋

isQuick(String):boolean, 判断输入的是否是快速移动指令

quick(String)快速移动

showAnswer()展示过关方法

reverse()旋转地图，让你第一关都会晕的功能，由于过于变态且不加分被舍弃

save(), load()存档读档

Judging Methods:

win\_judge()胜局判断

dead\_1()死局1判断

around(int, int):boolean根据某个箱子四周的情况判断箱子能否移动

dead\_2()死局2

reach(int, int, int[][])递归函数，确定人能到达的地图范围

Tool Methods

timer(long)计时

copymap(int[][]): int[][]复制地图

input(): String获取输入的指令

**遇到问题：**

1. IOHepler如何使用
2. 如何实现文件IO
3. 如何判断Mover脚下的是空地还是箱子目的地
4. 如何记录历史地图
5. 代码是面对过程的编程思想（当时没学到对象），如何改进
6. 推箱子好难过不了关怎么办
7. 很早以前写的程序，遇到的问题的忘记了，怎么办

**解决问题**

1. 曾尝试使用IOHelper，一小时后果断放弃，自己写IO。之后对Java的了解加深，自然就会了。
2. 使用Goolge和百度在网上搜索，最后了解到File类和FileWriter类，成功实现IO
3. 根据上一张地图里对应坐标的元素来判断，由此牵涉到问题4记录历史地图
4. 使用过三种方法记录历史地图：
   1. 使用一个二维int型数组记录地图。方法：每一个坐标都是0-9的个位数，而int型远不止能记录个位数。因此每当需要生成新地图时，将上一幅地图每个坐标处的数字乘以十，再加上新地图的数字，实现在一个二维int型数组里记录多幅地图。
      1. 优点：占用内存空间小，同时能实现少量步数的悔棋。
      2. 缺点：int型变量范围有限，记录地图的数量受到限制。
   2. 使用三维int型数组记录地图。
      1. 优点：简单有效，能够记录足够多的地图。
      2. 缺点：占用内存空间大，未作数据域封装。
   3. 使用对象记录地图，每个对象中有一个二维数组记录地图
      1. 优点：可以进行数据域封装，使程序更科学。

注：使用Stack类或者Arraylist类也可以很好地实现记录地图。

1. 重写程序。
2. 慢慢玩。。。或者找TA要答案。
3. 没办法呀，只好找几条充数，比如这条、上一条和上上条=。=

李逢双

13302010002