Lab2 制作地雷图

1. 主要算法设计

1.思路:首先读入命令行输入的地雷图的长度、宽度，以及拥有地雷的数量.并以此生成二维数组地雷图.地雷图的打印大致分四个步骤:创建规定大小的二维数组,随机生成规定数目地雷,非地雷处表示周围雷数量,打印。

2.创建数组:根据用户输入的地雷图数据,创建两个地雷图大小的数组,一个字符类型做地雷图(lable),一个布尔类型检测所处位置是否是地雷(state).

3.随机生成地雷:地雷坐标利用Math.random()分别生成行,列后给相应数组元素赋值字符\*.

4.非地雷处表示周围地雷数量:

(1)使用嵌套for循环遍历地雷图数组元素,if检测是否为地雷位置(state为真),如果是,跳出当前循环.

(2)初始化地雷计数器为0,嵌套for循环从左到右,从上到下遍历元素周围元素,若是地雷,则计数器加1.结束该嵌套循环后,将计数器的值转为字符赋值给相应数组元素.

5.打印:嵌套for循环顺序按格式打印出数组元素(地雷图).

二、边界问题的统计

统计周围地雷数量时,需要遍历以目标元素为中心的九宫格内数组元素,并判断是否为地雷.遍历时使用嵌套for循环实现对九宫格9个元素从左到右,从上到下的遍历,设置各个(行,列)for循环的结束条件为行/列数的最大值不超过地雷图大小且行/列循环次数不大于4.在各个for循环开始时设置if语句判断最小行/列数是否在地雷图之外,如果是,跳出本次循环.如此便可以解决边界元素周边地雷的统计问题.

三、对象和类

该Lab中按实验要求定义一个Board类,以下为UML类图

Board

- rows : int

- cols : int

- bombs : int

- bombLable : char[][]

- state : boolean[][]

+Board()

+setArgument(rows:int,cols:int,bombs:int):void

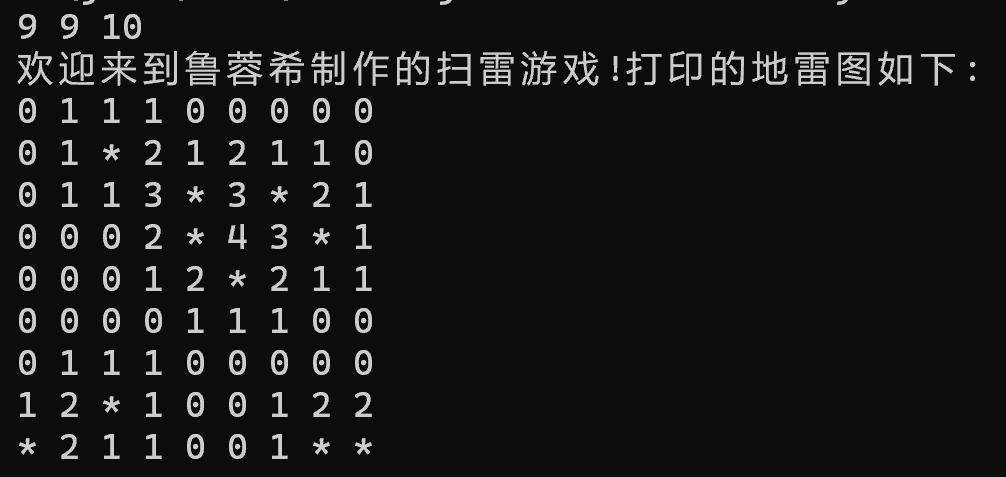
-creatArray():void

-randomBombs():void

-writeNumber():void

+printBombLable():void

四、运行效果截图



附录:

关键代码:

一、随机生成地雷位置的算法方案

**private** **void** randomBomb() {

**for**(**int** i = 0;i < bombs;i++) {

**int** bRow = (**int**)(Math.*random*() \* rows);

**int** bCol = (**int**)(Math.*random*() \* cols);

**if**(state[bRow][bCol])

i--;

bombLable[bRow][bCol] = '\*';

state[bRow][bCol] = **true**;

}

}

二、 地雷计数和边界判断的部分

private void writeNumber() {

for(int i = 0;i < rows;i++) {

for(int j = 0;j < cols;j++) {

if(state[i][j])

continue;

int bombCount = 0;

for(int r = -1;r + i <rows & r < 2;r++) {

if(r + i < 0)

continue;

for(int c = -1;c + j < cols & c < 2;c++) {

if(c + j < 0)

continue;

if(state[r + i][c + j])

bombCount++;

}

}

bombLable[i][j] = String.valueOf(bombCount).charAt(0);/\*数值转换为字符\*/

}

}

}