**lab2实验报告**

**实验内容分析：**

Lab2要求实现对于扫雷游戏地图的生成，其中重点在于地雷的随机布置以及每个位置周围地雷数的计数。除了主函数外，我定义了makeMap，printMap两个函数分别实现地图的生成，地图的打印。

**具体函数分析：**

**makeMap函数：**此函数为本次lab的核心，具体内容见附录图片。

首先，我使用Random生成两个在0-4之间的随机数，分别对应横纵坐标，同时加判断保证地雷的生成不能发生重复导致地雷数变少。以上循环指定的地雷数次。

其次，要判断除地雷外的每个位置的周围八个位置的地雷数来确定自身的数字。我循环遍历每个位置，排除地雷位置后，再通过两重最多八次的循环判断周围的地雷数，填入自身即可。

**printMap函数：**基本和lab1的print相同，按照行列顺序打印即可。

**实验难点：**

1. 生成地雷时，要保证地雷不能在重复的位置产生，防止地雷数减少。
2. 判断每个除地雷处的数字是，我使用了Math.min和Math.max函数来简化代码，这样既可以有效产生中心处，同时也照顾到了边界处的情况。

附录：

public static void main(String[] args)

{

int L,W,num;

Scanner input=new Scanner(System.in);

L=input.nextInt();

W=input.nextInt();

num=input.nextInt();

char [][]map=new char[L][W];

makeMap(map,num);

System.out.println("欢迎来到" +

"吴优\_22307130158" +

"制作的扫雷游戏！打印的地雷图如下");

printMap(map);

}

public static void makeMap(char [][]a,int num)

{

Random generator=new Random();

for(int i=0;i<num;)

{

int m1=generator.nextInt(a.length),m2= generator.nextInt(a[0].length);

if(a[m1][m2]!='\*')

{

a[m1][m2]='\*';

i++;

}

}

for(int i=0;i<a.length;i++)

{

for(int j=0;j<a[0].length;j++)

{

int val=0;

if(a[i][j]!='\*')

{

for(int k=Math.max(i-1,0);k<=Math.min(i+1,a.length-1);k++){

for(int t=Math.max(j-1,0);t<=Math.min(j+1,a[0].length-1);t++){

if(a[k][t]=='\*'){

val++;

}

}

}

a[i][j]=(char)(val+'0');

}

}

}

}

public static void printMap(char[][]a)

{

for(char[]arr:a)

{

for(char i:arr)

{

System.out.print(i+" ");

}

System.out.println();

}

}