

Lab7

TA:

- 14302010005@fudan.edu.cn 姜卓立
- 14302010040@fudan.edu.cn 武多才
- 14302010042@fudan.edu.cn 何培剑

1 摘要

1. 巩固并熟练运用数组
2. 学习方法的定义与使用
3. 综合使用方法，条件语句，循环语句，数组等知识实现“生命游戏”

2 方法

2.1 什么是方法？

- 方法是解决一类问题的步骤的有序组合
- 方法包含于类或对象中
- 方法在程序中被创建，在其他地方被引用

2.2 为什么使用方法？

- 使程序变得更简短而清晰。
- 有利于程序维护。
- 可以提高程序开发的效率。
- 提高了代码的重用性。

2.3 如何定义方法？

```
修饰符 返回值类型 方法名(参数类型 参数名){  
    ...  
    方法体  
    ...  
    return 返回值;  
}
```

例：

```
public static void main(String[] args){
    System.out.println("hello world");
}

public static int compare(int a, int b) {
    int result = a > b ? 1 : (a == b ? 0 : -1);
    return result;
}
```

2.4 命名规则

方法的命名规则同变量的命名规则一样：

- 必须以字母、'_'或'\$'开头
- 可以包括数字，但不能以它开头

2.5 命名规范

命名规范建议采用驼峰式命名法，一般第一个单词是动词，首字母小写

3 作业：生命游戏

3.1 介绍

生命游戏（Game of Life）是英国数学家约翰·何顿·康威在1970年发明的细胞自动机。它包括一个二维矩形世界，这个世界中的每个方格居住着一个活着的或死了的细胞。一个细胞在下一个时刻生死取决于相邻八个方格中活着的或死了的细胞的数量。如果相邻方格活着的细胞数量过多，这个细胞会因为资源匮乏而在下一个时刻死去；相反，如果周围活细胞过少，这个细胞会因太孤单而死去。

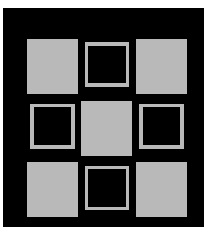
在这个世界中，对于任意细胞，规则如下：

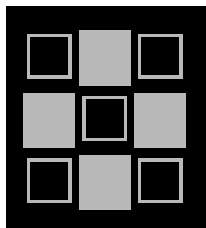
每个细胞有两种状态-存活或死亡，每个细胞与以自身为中心的周围八格细胞产生互动。（如图，黑色为存活，白色为死亡）

1. 当前细胞为存活状态时，当周围低于2个（不包含2个）存活细胞时，该细胞变成死亡状态。（模拟生命数量稀少）
2. 当前细胞为存活状态时，当周围有2个或3个存活细胞时，该细胞保持原样。
3. 当前细胞为存活状态时，当周围有3个以上的存活细胞时，该细胞变成死亡状态。（模拟生命数量过多）
4. 当前细胞为死亡状态时，当周围有3个存活细胞时，该细胞变成存活状态。（模拟繁殖）

可以把最初的细胞结构定义为种子，当所有在种子中的细胞同时被以上规则处理后，可以得到第一代细胞图。按规则继续处理当前的细胞图，可以得到下一代的细胞图，周而复始。

例：





3.2 作业要求

本次lab要求在已有的代码基础上,根据以下要求完成

1. 实现方法findLifedNum获取细胞 (x,y) 周围活细胞的个数
2. 实现方法transform产生下一代细胞结构

GameOfLife.java

```

import java.io.File;
import java.io.FileNotFoundException;
import java.util.Scanner;

public class GameOfLife {

    /**
     * 生命游戏入口
     *
     * @param cells 表示细胞图的二位数组
     */
    public static void start(int[][] cells) {
        Scanner scanner = new Scanner(System.in);
        int generation = 1;
        while (true) {
            System.out.println("generation:" + generation);
            printCells(cells);
            System.out.println("输入任意键产生下一代。。。");
            String input = scanner.nextLine();
            cells = transform(cells);
            generation++;
        }
    }

    /**
     * 统计cells[x][y]细胞周围活着的个数
     *
     * @param cells 细胞结构数组
     * @param x     横坐标
     * @param y     纵坐标
     * @return cells[x][y]细胞周围活着的个数
     */
    public static int findLifedNum(int[][] cells, int x, int y) {
        //code here
        return 0;
    }

    /**
     * 产生下一代
     *
     * @param cells 细胞结构数组
     * @return 新一代的细胞结构
     */
    public static int[][] transform(int[][] cells) {
        //code here
        return cells;
    }

    /**
     * 在控制台输出细胞结构
     *

```

```

    * @param cells 表示细胞图的二位数组
    */
    public static void printCells(int[][] cells) {
        for (int[] line : cells) {
            for (int cell : line) {
                System.out.print(cell == 0 ? "□" : "■");
            }
            System.out.println();
        }
    }

    public static void main(String[] args) throws FileNotFoundException {
        File file = new File(args[0]);
        Scanner scanner = new Scanner(file);
        int width = scanner.nextInt();
        int height = scanner.nextInt();
        int[][] cells = new int[width][height];
        for (int i = 0; i < width; i++) {
            for (int j = 0; j < height; j++) {
                cells[i][j] = scanner.nextInt();
            }
        }
        scanner.close();
        start(cells);
    }
}

```

3.3 Tips

1. 本次lab的细胞结构数据存放在文件1.txt、2.txt、3.txt，第一行两个数字分别表示细胞结构数组的行数与列数，从第二行开始的数组表示数组结构，'0'表示死细胞，'1'表示活细胞。请根据要求完成程序，并通过这3个文件的测试，使用以下格式加载测试文件：

java GameOfLife [Path]

例如：将程序使用javac编译后，运行命令 java GameOfLife 1.txt 表示运行第一个测试文件

2. 若使用cmd命令行运行程序，则测试文件需与运行的class文件同目录，若使用IntelliJ或Eclipse，则该文件需放在程序根目录下，即与.idea文件夹或.classpath文件同目录
3. 本次lab只需要完成方法findLifedNum与transform，其他方法无需改动

4 提交

1. 提交地址：ftp://10.132.141.33/classes/17/171 程序设计A(戴开宇)/WORK_UPLOAD/lab7/
2. 提交物：将java文件打包，命名格式为lab7_[学号]，例如：lab7_17302010001.zip
3. Deadline: 2017年11月12日23:59:59

5 声明

任何形式的作业都欢迎同学们相互讨论，但抄袭是严格禁止的。一旦发现抄袭行为，抄袭者和被抄袭者都以0分处理

