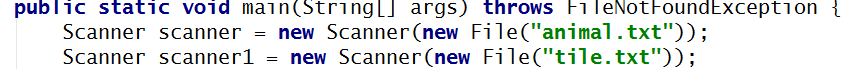
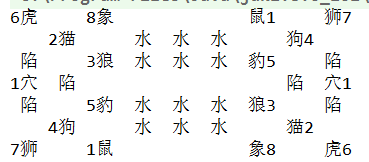
**斗兽棋设计文档**

**一．程序结构设计与分析：**

1. 读入和初始化地形地图、动物地图的方法：

读入地图的方法就是利用助教给的Scanner方法，然后再加上一句处理方法去解决找不到文件的报错

这样我们就将文本的内容读入，然后就要将对应数字转化为对应动物及地形。

这个时候我们可以用switch-case语句，也可以用字符串的形式将数字转化为对应文字。我采取了字符串对应方法，即建立一个一维字符串数组，然后运用for循环遍历数组，比如遇到数字1，就对应字符串第二位“1鼠 ”，再加上判断，将动物区分阵营（数字的左右），注意全半角，这样之后我们就将动物与地图读入的数字变成了字符串。接下来只需判断某地方是否有动物，有则输出动物，否则，输出地形。然后，我们就得到了这样的一个最终的地图。

1. 类的作用和设计思路：

我在这个程序里只建立了一个类，也就是public后的主类AnimalChess。

1. 关键变量的变量类型及作用（数组，类变量）：

我的程序中的关键变量大致有以下几个：

类变量：【1】两个字符串类型的二维数组，这两个数组定义在class里，main方法之外，便于整个程序调用。这两个作用是储存动物的位置和地形，animal数组是会随着游戏进行而改变其中的变量，而map始终保持不变。

【2】一个字符串类型的三维数组，这个数组也定义在class里，main方法之外，便于整个程序调用。这个的作用是储存棋盘的状态，以便悔棋和取消悔棋。当每次走棋时，animalHistory会储存当前棋盘的状态，当悔棋时，调用上一步的棋盘状态即可。

【3】两个int型的变量player和lastStep。player变量是用来记录当前步数以及该那个玩家行动的，若player是2的倍数，则左方移动，否则右方行动。同时每次移动后，player自增，用来在animalHistory数组中储存当前棋盘状态。lastStep在撤销悔棋时起作用，用来表示可以悔棋的最大步数。

关键变量：【4】两个int型变量m1，m2（这是最最关键的变量，移动全靠这两个！）m1和m2是用来定位所要移动的动物的，m1代表其行，m2代表其列，对动物进行移动就是对m1，m2操作。

【5】一个字符串类型的变量s。s是用户自己输入的字符串，根据s的内容，判断下一步操作该如何进行。

1. 主要的方法及功能（签名，参数，返回值，对变量的操作）：

【1】main方法：我的程序主要都在main方法执行，在main方法里进行了移动，重新开始，退出游戏，悔棋以及取消悔棋等操作。没有返回值。

【2】String[][] getAnimal()和String[][] getMap()方法：这两个方法是为了对类里的animal数组和map数组初始化，返回值是初始化后类里定义的animal数组map数组（二维字符串）。

【3】**int** assureRow(String m)和**int** assureColumn(String m)方法：这两个方法带的参数是一个字符串s，被调用时即用户输入的指令，目的是为了找到用户想要移动的动物的行与列，以便接下来移动的操作，返回值分别是是对应行与列的整数。

【4】void printLandscape()和void printRules()方法：这两个方法是void类型，目的是打印出完整合理的地图以及游戏规则。

【5】**boolean** isValidInput(String m)和**boolean** checkHaveAnimals()方法：这两个是boolean类型的方法，isValidInput是用来检验用户输入的移动指令是否符合要求；checkHaveAnimals是用来检验是否有一方的动物已被全部吃完。两个方法返回值都是布尔值。

【6】String[][] copyArray(String[][] array)方法：用来深度复制二维数组，以前lab7出现过，参数是String类型的二维数组，返回值是复制过的数组newArray。

1. 好处和坏处：我的程序结构好处在于基本上是正常第一感觉的想法，比较容易理解。但是一个比较明显的缺点在于用的方法比较少，基本上都在main方法里执行，看起来比较繁琐，重复的也比较多，太不简洁。
2. **编程中的问题：**
3. 最开始的时候按照助教给的方法读取地图然后switch-case输出对应文字，但是这样后面不方便阵营判断。
4. 输出地图之后我发现animal和map数组不能在其他方法里被调用，我当时没有想到为这两个数组建立一个方法。
5. 这次编程我之前最大的问题在于不会移动动物，不知道怎么找到对应的动物并且把这个行与列能传值到移动的操作里去。
6. 一开始不知道怎么保证轮流下且保证不出bug。

5.在编写规则的时候，一开始没有考虑到那么多限制，而且用if-else判断没有注意先后顺序，造成一些错误。

6.不知道无子可动应该怎么做，感觉我的思路十分繁琐，而且不能一一罗列。

7.悔棋一开始知道思路是什么，但是不知道怎么实现。

8.整体上而言，我对于整个程序一开始知道思路是什么，但是不知道怎么执行，不知道怎么合适地建方法传值，导致我一直停滞不前。

9.上传至github个人仓库的时候，我第一次没有用命令行直接上传，但是文件夹莫名被隐藏，不能上传整个project。

1. **解决办法：**
2. 自己想了很久，然后问了同学，尝试了四五种方法，发现可以用字符串将数字替代，这样后面可以判断阵营。
3. 经过同学的指点以及上课老师的提示，我发现可以将这两个数组定义在类里，这样就能被后面的方法调用啦，当然建立一个方法也是可以的，只是我不太会用方法。
4. 和同学讨论了好久，终于明白了原理，然后尝试建立方法，并且看了书上的例子，知道了怎么从方法返回值，然后完成了操作。
5. 经过井字棋这个简单的例子，我学会了continue和break这个办法，但是我觉得这个还有改进的地方，比如用三元操作符应该更加精简。

5.我仔细地阅读自己的程序，理清思路，调整顺序，分情况写好了规则。后来也有不少忽略规则限制的地方，但是都被一一解决。

6.询问做过的同学，但是还是无法理解，最后放弃了做这个要求。

7.lab7提供了一个方法，写起来十分简便。

8.经过阅读书上的例子以及井字棋的示例和Bing的棋类游戏思路，我渐渐有了大体的构思，然后再不断完善规则等细节。

9.询问李逢双助教并阅读教程后，还是采取命令行方法，成功导入个人仓库。