概要:1个类，4个方法

方法:

1. main:读取文件内容并初始化为矩阵，用二维数组存储，并调用其他函数输出最后结果
2. Calculate: 计算两个矩阵相乘的方法，返回一个二维数组
3. Recursion:递归n次幂方法
4. Traversal:遍历二维数组并输出

package Matrix;

import java.util.\*;

import java.io.\*;

public class Matrix {

static int order;

static int result=0;

static int R1[][];

static int R0[][];

static HashMap<Character,Integer>hm;

public static void main(String args[]){

//用hashmap将字母映射成数字

hm=new HashMap<>();

hm.put('a', 0);

hm.put('b', 1);

hm.put('c', 2);

hm.put('d', 3);

//读取文件内容并初始化为矩阵，用二维数组存储

try{

InputStreamReader read=new InputStreamReader(

new FileInputStream(new File("D:/java\_project/pokedo/SelfLearning/Matrix/test.txt")));

BufferedReader reader = new BufferedReader(read);

order=reader.readLine().split(",").length;

System.out.println(order);

//初始化R1矩阵

R1=new int[order][order];

String line;

while((line=reader.readLine())!=null){

int i=hm.get(line.charAt(0));

int j=hm.get(line.charAt(1));

System.out.print(i);

System.out.println(j);

R1[i][j]=1;

}

//初始化R0矩阵

R0=new int[order][order];

for(int i=0;i<order;i++)

R0[i][i]=1;

}

catch(Exception e)

{e.printStackTrace();}

//输出R1矩阵的n次幂的结果

Traversal(R1);

System.out.println("\n5次幂:");

Traversal(Recursion(5,R1));

}

//计算两个矩阵相乘的方法，返回一个二维数组

public static int[][] Calculate(int Rk[][],int Rb[][])

{

int Rs[][]=new int[order][order];

for(int t=0;t<order;t++){

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

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

result+=Rk[t][j]\*Rb[j][i];

if(j==order-1){

if(result!=0)

Rs[t][i]=1;

else

Rs[t][i]=0;

result=0;

}

}

}

}

return Rs;

}

//递归n次幂方法

public static int[][] Recursion(int n,int Rn[][]){

if(n==1){

return Calculate(R1,R0);

}

else if(n==2){

return Calculate(R1,R1);

}

else{

return Calculate(Recursion(--n,R1),R1);

}

}

//遍历二维数组并输出

public static void Traversal(int R[][]){

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

{

System.out.println();

for(int j=0;j<R.length;j++)

{

System.out.print(R[i][j]+" ");

}

}

}

}

