

Отчет

По дискретной математике

Тема: раскладка поля фигурами домино

Код:

```
1 package domino;
2 import domino.*;
3 import java.util.Scanner;
4 /**
5  *
6  * @author Kravchenko Alexey
7  */
8
9 import java.util.Scanner;
10
11 /**
12  * Created by StormSeadle on 28.05.2017.
13  */
14 public class Domino {
15     //функция вычисляющая количество способов разложения фигурками домино
16     public static int sposob(int[][] matr, int sum, int a, int b)
17     {
18         boolean f = true;
19         boolean f1 = true;
20         //Всего существует два положения у фигурки домино, поэтому создается две дополнительные матрицы
21         int[][] matr1 = new int[a][b];
22         int[][] matr2 = new int[a][b];
23         for(int i=0; i<a; i++){
24             for(int j=0; j<b; j++){
25                 matr1[i][j] = matr[i][j];
26                 matr2[i][j] = matr[i][j];
27             }
28         }
29         for(int i=0; i<a; i++){
30             for(int j=0; j<b; j++){
31                 //Первый элемент общий при любом расположении
32                 if (matr[i][j]==1){
33                     matr1[i][j]=0;
34                     matr2[i][j]=0;
35                     //Проверка на возможно расположения костяшки(выхлз за границу)
36                     if((j<b-1) && (matr2[i][j+1] == 0)){f=false;}
37                     if((i<a-1) && (matr1[i+1][j]==0)){f1=false;}
38                     if(i<a-1){matr1[i+1][j] = 0;}
39                     if(j<b-1){matr2[i][j+1] = 0;}
40                     if(i>=a-1)f1=false;
41                     if(j>=b-1) f=false;
```

```

25         matr1[i][j] = matr[i][j];
26         matr2[i][j] = matr[i][j];
27     }
28 }
29 for(int i=0; i<a; i++){
30     for(int j=0; j<b; j++){
31         //Первый элемент общий при любом расположении
32         if (matr[i][j]==1){
33             matr1[i][j]=0;
34             matr2[i][j]=0;
35             //Проверка на возможно расположения костяшки(выход за границу)
36             if((j<b-1) && (matr2[i][j+1] == 0)){f=false;}
37             if((i<a-1) && (matr1[i+1][j]==0)){f1=false;}
38             if(i<a-1){matr1[i+1][j] = 0;}
39             if(j<b-1){matr2[i][j+1] = 0;}
40             if(i>=a-1)f1=false;
41             if(j>=b-1) f=false;
42             //При возможности одного из способов вызывается соответствующая функция, где одна из костяшек уже однозначна
43             if(f && f1) return sposob(matr1, sum-2, a, b) + sposob(matr2, sum-2, a, b);
44             if(f && !f1) return sposob(matr2, sum-2, a, b);
45             if(!f && f1) return sposob(matr1, sum-2, a, b);
46         }
47     }
48 }
49 //Если остаются две свободные клетки то способ расположения всего один
50 if(sum==2){
51     return 1;
52 }else return 1;
53 }
54 public static void main(String[] args) {
55     int a, b;
56     int white=1,black=0;
57     Scanner in = new Scanner(System.in);
58     System.out.println("Введите количество строк");
59     a=in.nextInt();
60     System.out.println("Введите количество столбцов");
61     b=in.nextInt();
62     int[][] c = new int[a][b];
63     char[][] k = new char[a][b];
64     int sum=0;
65     System.out.println("Введите двумерным массивом, где 1 означают точки фигуры, а 0 - дырки");

```

```

67     for (int i=0; i<a; i++)
68     {
69         for(int j=0; j<b; j++)
70         {
71             c[i][j]=in.nextInt();
72             if(c[i][j]==1)sum+=1;
73         }
74     }
75     //Если количество клеток нечетное, то невозможно разложить поле, т.к. костяшки 2x1
76     if((a*b)%2!=0)
77     {
78         System.out.println("Нельзя разложить фигуру домино");
79         System.exit(0);
80     }
81     //Чтобы проверить можно ли разложить поле фигурками домино, можно раскрасить поле в шахматном цвете
82     //и если кол-во черных = кол-во белых, то такое поле можно разложить
83     k[0][0]='w';
84     for(int i=0; i<a; i++)
85     {
86         for(int j=0; j<b; j++)
87         {
88             boolean f = false;
89             if(i>0){if(k[i-1][j]=='w')f=true;}
90             if(j>0){if(k[i][j-1]=='w')f=true;}
91             if(!f)
92             {
93                 k[i][j]='b';
94                 black+=1;
95             }
96             f = false;
97             if(i>0){if(k[i-1][j]=='b')f=true;}
98             if(j>0){if(k[i][j-1]=='b')f=true;}
99             if(!f)
100             {
101                 k[i][j]='w';
102                 white+=1;
103             }
104         }
105     }
106     //Проверка на то, стоит ли клетка одна
107     for (int i=0; i<a; i++){
108         for(int j=0; j<b; j++)
109         {
110             int f=0;
111
112             if(i>0){
113                 if(c[i-1][j]==0){
114                     f+=1;

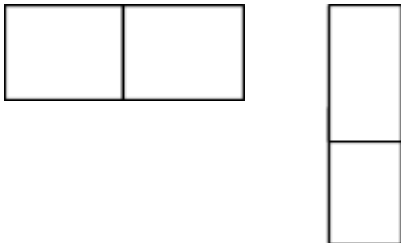
```

```
112         if(i>0){
113             if(c[i-1][j]==0){
114                 f+=1;
115             }
116         }
117         if(i==0){
118             f+=1;
119         }
120         if(i<a-1){
121             if(c[i+1][j]==0){
122                 f+=1;
123             }
124         }
125         if(i==a-1){
126             f+=1;
127         }
128         if(j==0){
129             f+=1;
130         }
131         if(j<b-1){
132             if(c[i][j+1]==0){
133                 f+=1;
134             }
135         }
136         if(j==b){
137             f+=1;
138         }
139         if((f==4) && (c[i][j]==1))
140         {
141             System.out.println("Нельзя разложить фигурку домино");
142             System.exit(0);
143         }
144     }
145 }
146 //Проверка на количество черных и белых клеток
147 if(white!=black)
148 {
149     System.out.println("Нельзя разложить фигурку домино");
150     System.exit(0);
151 }else
152 {
153     System.out.println("Можно разложить фигурку домино");
154 }
155 //Вызов функции
156 System.out.println(sposob(c, sum, a, b));
157 }
158 }
159 }
```

Общий алгоритм

Алгоритм работает методом полного перебора при помощи рекуррентной функции.

Всего два возможных расположения костяшки домино:



Программа пробует поставить костяшки этими способами и так пока не останется два свободных места на поле. Значит это единственный вариант. И все результаты суммируются.

Проверка возможности разложения костяшек определяется с помощью шахматной раскраски поля. Если количество черных занятых клеток и количество белых равны, то на таком поле можно разложить костяшки домино.