

```
1 package tests;
2
3 import static org.junit.Assert.*;
4
5 import java.util.Set;
6
7 import org.junit.Before;
8 import org.junit.Test;
9
10 import experiment.BoardCell;
11 import experiment.IntBoard;
12
13 public class IntBoardTests
14 {
15     IntBoard board;
16     @Before
17     public void beforeAll()
18     {
19         board = new IntBoard();
20     }
21
22     @Test
23     public void testAdjacency0()
24     {
25         beforeAll();
26         // board = new IntBoard();
27         //IntBoard board = new IntBoard();
28
29
30         BoardCell cell = board.getCell(0,0);
31         System.out.println(cell);
32         Set<BoardCell> testList = board.getAdjList(cell);
33         //assertTrue(testList.contains(board.getCell(1, 0)));
34         //assertTrue(testList.contains(board.getCell(0, 1)));
35
36         assertEquals(2, testList.size());
37     }
38
39     @Test
40     public void testAdjacency1()
41     {
42         beforeAll();
43         BoardCell cell = board.getCell(3,3);
44         Set<BoardCell> testList = board.getAdjList(cell);
45         assertTrue(testList.contains(board.getCell(3, 2)));
46         assertTrue(testList.contains(board.getCell(2, 3)));
47         assertEquals(2, testList.size());
48     }
49
50     @Test
51     public void testAdjacency2()
52     {
53         beforeAll();
54         BoardCell cell = board.getCell(1,3);
55         Set<BoardCell> testList = board.getAdjList(cell);
56         assertTrue(testList.contains(board.getCell(2, 3)));
57         assertTrue(testList.contains(board.getCell(1, 2)));
58         assertTrue(testList.contains(board.getCell(0, 3)));
59     }
60 }
```

```
61
62     assertEquals(3, testList.size());
63
64 }
65 @Test
66 public void testAdjacency3()
67 {
68     beforeAll();
69     BoardCell cell = board.getCell(3,0);
70     Set<BoardCell> testList = board.getAdjList(cell);
71     assertTrue(testList.contains(board.getCell(2, 0)));
72     assertTrue(testList.contains(board.getCell(3, 1)));
73     assertEquals(2, testList.size());
74
75 }
76 @Test
77 public void testAdjacency4()
78 {
79     beforeAll();
80     BoardCell cell = board.getCell(1,1);
81     Set<BoardCell> testList = board.getAdjList(cell);
82     assertTrue(testList.contains(board.getCell(0, 1)));
83     assertTrue(testList.contains(board.getCell(2, 1)));
84     assertTrue(testList.contains(board.getCell(1, 2)));
85     assertTrue(testList.contains(board.getCell(1, 0)));
86
87
88     assertEquals(4, testList.size());
89
90 }
91 @Test
92 public void testAdjacency5()
93 {
94     beforeAll();
95     BoardCell cell = board.getCell(2,2);
96     Set<BoardCell> testList = board.getAdjList(cell);
97     assertTrue(testList.contains(board.getCell(2, 1)));
98     assertTrue(testList.contains(board.getCell(3, 2)));
99     assertTrue(testList.contains(board.getCell(2, 3)));
100    assertTrue(testList.contains(board.getCell(1, 2)));
101
102
103    assertEquals(4, testList.size());
104
105 }
106 ////////////////////////////////////////////////////
107 @Test
108 public void testTargets0_0_3()
109 {
110     BoardCell cell = board.getCell(0, 0);
111     board.calcTargets(cell, 3);
112     Set<BoardCell> targets = board.getTargets();
113     // assertEquals(6, targets.size());
114     assertTrue(targets.contains(board.getCell(3, 0)));
115     assertTrue(targets.contains(board.getCell(2, 1)));
116     assertTrue(targets.contains(board.getCell(0, 1)));
117     assertTrue(targets.contains(board.getCell(1, 2)));
118     assertTrue(targets.contains(board.getCell(0, 3)));
119     assertTrue(targets.contains(board.getCell(1, 0)));
120 }
```

```
121 @Test
122 public void testTargets0_3_2()
123 {
124     BoardCell cell = board.getCell(0, 3);
125     board.calcTargets(cell, 2);
126     Set<BoardCell> targets = board.getTargets();
127     // assertEquals(4, targets.size());
128     assertTrue(targets.contains(board.getCell(2, 3)));
129     assertTrue(targets.contains(board.getCell(0, 1)));
130     assertTrue(targets.contains(board.getCell(1, 2)));
131     assertTrue(targets.contains(board.getCell(1, 2)));
132
133 }
134 @Test
135 public void testTargets3_1_2()
136 {
137     BoardCell cell = board.getCell(3, 1);
138     board.calcTargets(cell, 2);
139     Set<BoardCell> targets = board.getTargets();
140     // for (BoardCell target : targets)
141     // {
142     //     System.out.println(target.column + " " +target.row);
143     // }
144     // System.out.println("Balls " + targets);
145     assertEquals(4, targets.size());
146     assertTrue(targets.contains(board.getCell(3, 3)));
147     assertTrue(targets.contains(board.getCell(2, 2)));
148     assertTrue(targets.contains(board.getCell(1, 1)));
149     assertTrue(targets.contains(board.getCell(2, 0)));
150
151 }
152 @Test
153 public void testTargets1_0_3()
154 {
155     BoardCell cell = board.getCell(1, 0);
156     board.calcTargets(cell, 3);
157     Set<BoardCell> targets = board.getTargets();
158     //assertEquals(6, targets.size());
159     assertTrue(targets.contains(board.getCell(3, 1)));
160     assertTrue(targets.contains(board.getCell(2, 2)));
161     assertTrue(targets.contains(board.getCell(1, 3)));
162     assertTrue(targets.contains(board.getCell(0, 2)));
163     assertTrue(targets.contains(board.getCell(0, 0)));
164     assertTrue(targets.contains(board.getCell(1, 1)));
165
166 }
167
168
169 @Test
170 public void testTargets2_3_2()
171 {
172     BoardCell cell = board.getCell(2, 3);
173     board.calcTargets(cell, 2);
174     Set<BoardCell> targets = board.getTargets();
175     // assertEquals(3, targets.size());
176     assertTrue(targets.contains(board.getCell(3, 2)));
177     assertTrue(targets.contains(board.getCell(2, 1)));
178     assertTrue(targets.contains(board.getCell(1, 2)));
179     assertTrue(targets.contains(board.getCell(0, 3)));
180
```

```
181     }
182
183     @Test
184     public void testTargets2_1_2()
185     {
186         BoardCell cell = board.getCell(2, 1);
187         board.calcTargets(cell, 2);
188         Set<BoardCell> targets = board.getTargets();
189         //assertEquals(6, targets.size());
190         assertTrue(targets.contains(board.getCell(3, 2)));
191         assertTrue(targets.contains(board.getCell(3, 0)));
192         assertTrue(targets.contains(board.getCell(0, 1)));
193         assertTrue(targets.contains(board.getCell(1, 0)));
194         assertTrue(targets.contains(board.getCell(1, 2)));
195         assertTrue(targets.contains(board.getCell(2, 3)));
196
197     }
198 }
199
```