

HW1_이주성_32183520

Board.java

- 기존에는 3x3 Board만 가능했지만 NxN Board가 가능하도록 구현했다.
 - `int size` 를 파라미터로 받는 생성자
 - `List<Integer> spots` 를 파라미터로 받는 생성자
- size를 구하는 size() 메서드 구현
- Board의 값을 구하는 valueAt 구현
 - index로 구하는 메서드
 - indices로 값 리스트를 구하는 메서드

BookShaper.java

- setBoard로 Board 지정
- row를 구하는 메서드

```
for (int i = (rowIndex - 1) * size; i < rowIndex * size; i++) {  
    rowValues.add(board.valueAt(i));  
}
```

- column을 구하는 메서드

```
for (int i = columnIndex - 1; i < size * size; i+=size) {  
    columnValues.add(board.valueAt(i));  
}
```

- diagonal()

```
for (int i = 0; i < size * size; i += size + 1) {
    diagonal.add(board.valueAt(i));
}
```

- transpose()

```
public Board transpose() {
    List<Integer> tt = new ArrayList<>();
    int size = board.size();

    for (int i = 0; i < size; i++) {
        for (int j = i; j < size * size; j += size) {
            tt.add(board.valueAt(j));
        }
    }

    return new Board(tt);
}
```

출력 메서드

- IBoardDisplay 인터페이스 사용

```
public interface IBoardDisplay {
    void display(Board board);
}
```

- BoardDisplay - IBoardDisplay 구현

- “|” 구분선으로 보기 좋게 출력

```
public class BoardDisplay implements IBoardDisplay {
    @Override
    public void display(Board board) {
        int size = board.size();

        for (int row = 0; row < size; row++) {
            for (int col = 0; col < size; col++) {
                int index = row * size + col;
                if (col == size - 1) {
                    System.out.print(board.valueAt(index));
                }
            }
        }
    }
}
```

```
        } else {  
            System.out.print(board.valueAt(index) + " | ");  
        }  
    }  
    System.out.println();  
}  
}  
}
```

결과

```

board [ 2 ] start
1 | 2
3 | 4

row#1 [1, 2]
row#2 [3, 4]

column#1 [1, 3]
column#2 [2, 4]

diagonal [1, 4]

transpose
1 | 3
2 | 4
=====end=====

board [ 3 ] start
1 | 2 | 3
4 | 5 | 6
7 | 8 | 9

row#1 [1, 2, 3]
row#2 [4, 5, 6]
row#3 [7, 8, 9]

column#1 [1, 4, 7]
column#2 [2, 5, 8]
column#3 [3, 6, 9]

diagonal [1, 5, 9]

transpose
1 | 4 | 7
2 | 5 | 8
3 | 6 | 9
=====end=====

```

```
board [ 4 ] start
1 | 2 | 3 | 4
5 | 6 | 7 | 8
9 | 10 | 11 | 12
13 | 14 | 15 | 16

row#1 [1, 2, 3, 4]
row#2 [5, 6, 7, 8]
row#3 [9, 10, 11, 12]
row#4 [13, 14, 15, 16]

column#1 [1, 5, 9, 13]
column#2 [2, 6, 10, 14]
column#3 [3, 7, 11, 15]
column#4 [4, 8, 12, 16]

diagonal [1, 6, 11, 16]

transpose
1 | 5 | 9 | 13
2 | 6 | 10 | 14
3 | 7 | 11 | 15
4 | 8 | 12 | 16
=====end=====
```

```
board [ 5 ] start
1 | 2 | 3 | 4 | 5
6 | 7 | 8 | 9 | 10
11 | 12 | 13 | 14 | 15
16 | 17 | 18 | 19 | 20
21 | 22 | 23 | 24 | 25

row#1 [1, 2, 3, 4, 5]
row#2 [6, 7, 8, 9, 10]
row#3 [11, 12, 13, 14, 15]
row#4 [16, 17, 18, 19, 20]
row#5 [21, 22, 23, 24, 25]

column#1 [1, 6, 11, 16, 21]
column#2 [2, 7, 12, 17, 22]
column#3 [3, 8, 13, 18, 23]
column#4 [4, 9, 14, 19, 24]
column#5 [5, 10, 15, 20, 25]

diagonal [1, 7, 13, 19, 25]

transpose
1 | 6 | 11 | 16 | 21
2 | 7 | 12 | 17 | 22
3 | 8 | 13 | 18 | 23
4 | 9 | 14 | 19 | 24
5 | 10 | 15 | 20 | 25
=====end=====
```