

```

1 package Project3;
2
3 /*****
4  * CIS 163 Section 01
5  * Project 3: Chess Game
6  * Bishop Class
7  *
8  * This class represents a ChessPiece that is a bishop.
9  *
10 * @author George Fayette
11 * @version 3/23/2019
12 *****/
13 public class Bishop extends ChessPiece {
14
15     /*****
16      * Public constructor sets player to parameter value.
17      * @param player The player type.
18      *****/
19     public Bishop(Player player) {
20         super(player);
21     }
22
23     /*****
24      * Public String, returns the ChessPiece type.
25      * @return A string representing the ChessPiece type.
26      *****/
27     public String type() {
28         return "Bishop";
29     }
30
31     /*****
32      * Public boolean, returns true if the move is valid.
33      * @param move The move that is being checked.
34      * @param board The array of IChessPieces that is being checked.
35      * @return True if the move is valid.
36      *****/
37     public boolean isValidMove(Move move, IChessPiece[][] board) {
38         boolean valid = true;
39
40         if (!super.isValidMove(move, board)) {
41             valid = false;
42         } else {
43             int vDistance = move.toRow - move.fromRow;
44             int hDistance = move.toColumn - move.fromColumn;
45
46             if (Math.abs(vDistance) - Math.abs(hDistance) != 0) {
47                 valid = false;
48             } else {
49                 int absDistance = Math.abs(vDistance);
50                 if (vDistance > 0 && hDistance > 0) {
51                     for (int i = 1; i < absDistance; ++i) {
52                         if (board[move.fromRow + i][move.fromColumn +
53                             i] != null) {
54                             valid = false;
55                         }
56                     }
57                 } else if (vDistance > 0 && hDistance < 0) {
58                     for (int i = 1; i < absDistance; ++i) {
59                         if (board[move.fromRow + i][move.fromColumn -
60                             i] != null) {

```

```
61         valid = false;
62     }
63 }
64 } else if (vDistance < 0 && hDistance > 0) {
65     for (int i = 1; i < absDistance; ++i) {
66         if (board[move.fromRow - i][move.fromColumn +
67             i] != null) {
68             valid = false;
69         }
70     }
71 } else if (vDistance < 0 && hDistance < 0) {
72     for (int i = 1; i < absDistance; ++i) {
73         if (board[move.fromRow - i][move.fromColumn -
74             i] != null) {
75             valid = false;
76         }
77     }
78 }
79 }
80 }
81 return valid;
82 }
83 }
```