

```

1 package Project3;
2
3 /*****
4  * CIS 163 Section 01
5  * Project 3: Chess Game
6  * Pawn Class
7  *
8  * This class represents a ChessPiece that is a pawn.
9  *
10 * @author George Fayette
11 * @version 3/23/2019
12 *****/
13 public class Pawn extends ChessPiece {
14
15     /**
16      * Public boolean representing whether or not the pawn has moved.
17      */
18     public boolean firstMove;
19
20     /**
21      * Public constructor sets player to parameter value.
22      * @param player The player type.
23      *****/
24     public Pawn(Player player) {
25         super(player);
26         firstMove = true;
27     }
28
29     /**
30      * Public String, returns the ChessPiece type.
31      * @return A string representing the ChessPiece type.
32      *****/
33     public String type() {
34         return "Pawn";
35     }
36
37     /**
38      * Public boolean, returns true if the move is valid.
39      * @param move The move that is being checked.
40      * @param board The array of IChessPieces that is being checked.
41      * @return True if the move is valid.
42      *****/
43     public boolean isValidMove(Move move, IChessPiece[][] board) {
44         boolean valid = true;
45         if (!super.isValidMove(move, board)) {
46             valid = false;
47         } else {
48
49             IChessPiece moveFrom = board[move.fromRow][move.fromColumn];
50             IChessPiece moveTo = board[move.toRow][move.toColumn];
51             int vDistance = move.toRow - move.fromRow;
52             int hDistance = move.toColumn - move.fromColumn;
53             int maxDistance = 1;
54
55             if (firstMove) {
56                 maxDistance = 2;
57             }
58
59             if (moveFrom.player() == Player.WHITE) {
60                 if (hDistance == 0) {

```

```
61         if (vDistance < -maxDistance || vDistance > 0) {
62             valid = false;
63         }
64         for (int i = move.fromRow - 1; i >= move.toRow;
65             --i) {
66             if (board[i][move.fromColumn] != null) {
67                 valid = false;
68             }
69         }
70     } else if ((hDistance != 1 && hDistance != -1) ||
71         vDistance != -1 || moveTo == null) {
72         valid = false;
73     }
74 }
75 } else if (moveFrom.player() == Player.BLACK) {
76     if (hDistance == 0) {
77         if (vDistance > maxDistance || vDistance < 0) {
78             valid = false;
79         }
80         for (int i = move.fromRow + 1; i <= move.toRow;
81             ++i) {
82             if (board[i][move.fromColumn] != null) {
83                 valid = false;
84             }
85         }
86     } else if ((hDistance != 1 && hDistance != -1) ||
87         vDistance != 1 || moveTo == null) {
88         valid = false;
89     }
90 }
91 }
92 }
93 return valid;
94 }
95 }
```