```java
 1  package Project3;
 2
 3  /*******************************************************************
 4   * CIS 163 Section 01
 5   * Project 3: Chess Game
 6   * King Class
 7   *
 8   * This class represents a ChessPiece that is a king.
 9   *
10   * @author George Fayette
11   * @version 3/23/2019
12   *******************************************************************/
13  public class King extends ChessPiece {
14
15      /**
16       * Public boolean representing whether or not the king 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 King(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 "King";
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
46          if (!super.isValidMove(move, board)) {
47              valid = false;
48          } else {
49              int vDistance = move.toRow - move.fromRow;
50              int hDistance = move.toColumn - move.fromColumn;
51
52              if (vDistance < 0) {
53                  vDistance *= -1;
54              }
55              if (hDistance < 0) {
56                  hDistance *= -1;
57              }
58
59              if (vDistance > 1 || hDistance > 1) {
60                  valid = false;
```

```
61                }
62            }
63         return valid;
64     }
65 }
```