chess.java

```java
/**
 * An input of 2 index positions in a 2d Array of size 8x8 will be taken. The program
 * should then check which chess piece can move from the first index to the second in 1 move.
 * The chess pieces move as follows:
 * King - 1 space in any direction
 * Queen - any number of spaces in forward, backward, left, right and diagonal directions
 * Bishop - any number of spaces in diagonal directions
 * Knight - 2 spaces in forward, backward, left or right directions followed by 1 space perpendicular to it.
 * Rook - any number of spaces in forward, backward, left and right directions
 * Pawn - 1 space in the forward direction only.
 */

import java.util.Scanner;

public class chess {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter start indexes");
        int x1 = sc.nextInt();
        int y1 = sc.nextInt();
        System.out.println("Enter end indexes");
        int x2 = sc.nextInt();
        int y2 = sc.nextInt();
        int flag = 0;
        sc.close();
        // Program is easily solved using distance formula between two points
        // The distance formula in geometry is ((y2-y1)^2 + (x2-x1)^2)^0.5
        // The right distance needed to attack a piece is constant so just needs to be checked once.
        double distance = Math.sqrt(Math.pow((y2 - y1), 2) + Math.pow((x2 - x1), 2));
        if (distance == 1 || distance == Math.sqrt(2)) {
            System.out.println("King");
            flag = 1;
        }
        if (distance % Math.sqrt(2) == 0) {
            System.out.println("Bishop");
            flag = 1;
        }
        if (x1 == x2 || y1 == y2) {
            System.out.println("Rook");
            flag = 1;
        }
        if (distance == Math.sqrt(5)) {
            System.out.println("Horsey!");
        }
        if (y1 == y2 && (x2 + 1 == x1)) {
            System.out.println("Pawn");
            flag = 1;
        }
        if (flag == 1) {
            System.out.println("Queen");
        }
    }
}
/**
 * Variable        Data        Table
 * x1, y1          int         Store coordinates of first point on chess board
 * x2, y2          int         Store coordinates of second point on chess board
 * distance        int         Calculate and store distance between the points
 * flag            int         Check for queen
 */
```