```java
import java.io.*;
import java.util.*;

/**
 * This class models a Binary Question Tree.
 *
 *  It first initialize a "computer" Answer Node as its root.
 *
 *  If previous game data exist, user can have the tree
 *   read that data and load that game
 *  When a game is finished, user can have the tree
 *   write down its structure
 *  When a game is played, the tree expands per lost game
 *   using data given by user
 *
 * <ul>
 * <li> Name: QuestionTree.java
 * <li> Description: Question Tree
 * <li> Class: Java 145
 * <li> Instructor: Ken Hang && Janet Ash
 * <li> Date: March 16 2015
 * </ul>
 * @author  Hai H Nguyen (Bill)
 * @version Winter 2015
 */
public class QuestionTree {
    private QuestionNode root;

    private Scanner user;

    /**
     * Constructor which initialize the QuestionNode
     * and the user Scanner
     */
    public QuestionTree(){
        root = new QuestionNode("computer");

        user = new Scanner(System.in);
    }

    /**
     * Play a complete guessing game
     * and rebuild the tree
     * as necessary
     */
    public void askQuestions() {
        root = playedNode(root);
    }

    private QuestionNode playedNode(QuestionNode node){
        if (node.isAnswer()) { // If a node is an Answer node
            if (yesTo ("Would your object happen to be " + node + "?")) {
                debugLog("Great, I got it right!\n"); // If user accepts the answer
            } else { // Else the machine learn a new question node
                node = learnedNode(node);
            }
        } else { // If node is a Question node
            if (yesTo (node.toString())) {
                node.yes = playedNode(node.yes);
            } else {
                node.no = playedNode(node.no);
            }
        }
        return node;
    }

    private QuestionNode learnedNode(QuestionNode node){
        debugLog("What is the name of your object? ");
        // Make new answer node for the new object
        QuestionNode newNode = new QuestionNode (user.nextLine());

        debugLog("Please give me a yes/no question that\n" +
                  "distinguishes between your object\n" +
                  "and mine--> ");
        // Get user's question
        String query = user.nextLine();
        // Make new Question Node per user's command
        return yesTo("And what is the answer for your object?") ?
                new QuestionNode ( query, newNode, node ):
                new QuestionNode ( query, node, newNode );
    }

    /**
     * Invokes the writeNodes Recursion and prints
     * the Question Tree into the output Stream
     * @param output        Output object to Stream data
     */
    public void write(PrintStream output){
        write(output, root);
    }

    private void write(PrintStream out, QuestionNode node){
        if (node.isAnswer()){
            out.print("A:\n" + node);
        } else {
            out.println("Q:\n" + node);

            write(out, node.yes);

            out.println();
            // Avoid new line at the end
            write(out, node.no);
        }
    }

    /**
     * Invokes the readNode recursion and builds
     * the Question Tree in a binary fashion
     * @param input         Scanner Object to get Data
     */
    public void read (Scanner input){
        root = readNode(input);
    }

    private QuestionNode readNode(Scanner input){
        QuestionNode node = null; // Initialise a null node.

        if (input.hasNextLine()){
            if (input.nextLine().startsWith("A:")){// Create Answer node
                node = new QuestionNode(input.nextLine());
            } else { // Else Create Question node
                node = new QuestionNode(input.nextLine(), readNode(input), readNode(input));
            }
        }

        return node;
    }

    /**
     * Asks the user a question, forcing an answer of "y " or "n";
     * @param prompt    Message to print
     * @return          True if the answer was yes, false otherwise
     */
    public boolean yesTo (String prompt){
        debugLog(prompt + " (y/n)? ");

        String response = user.nextLine().trim().toLowerCase();

        if (!response.equals("y") && !response.equals("n")) {
            debugLog ("Please answer y or n.\n");
            // U know Recursion? Slow Stack, but No Loop...
            return yesTo(prompt);
        }

        return response.equals("y");
    }

    private void debugLog(Object o){
        if(o!= null) {
            System.out.print(o.toString());
        }
    }
}//IS29
```