**Question:**

WAP in any high level language to implement the animal game.

**Background**

The animal guessing game is a program that identifies the animal which is user is thinking by asking a series of question. In the case the program cannot provide the animal name it asks the user what the animal was and a question that is pretty close to the animal so that next time the game is played the animal is guessed correctly.

This game is implemented using a binary tree. The expected answers are searched in the leaves of the tree. The leaf nodes are the final answers (animal names) and the non-leaf nodes are the question asked to the users. A new question is added as a non-leaf node and the answers yes or no are the two child of this node.

**Methodology**

The program is written in java and linked list data structure was used to implement the binary tree i.e. node. The node consists of the head node which is the question and two leaf nodes which may consists of further question if they are non-leaf node or animal name in case they are leaf node.

Example

[‘Does your animal fly?’, ‘Eagle’, ‘Frog’]

This is the list in a case the question has final answers as the nodes.

[‘Does your animal fly?’, ‘Eagle’, ‘[‘Does your animal swim?’, ‘Frog’, ‘Kangaroo’]

’]

List in a case the question has the answer ‘no’ that is a followed by further questions.

**Program**

import java.io.\*;

import java.util.Scanner;

public class AnimalGuess {

private static class Node implements Serializable {

public String question;

public Node left;

public Node right;

public Node(String question) {

this.question = question;

}

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

Node n = null;

try {

FileInputStream fis = new FileInputStream("tree.ser");

ObjectInputStream ois = new ObjectInputStream(fis);

n = (Node) ois.readObject();

ois.close();

} catch (Exception e) {

}

if (n == null) {

n = new Node("Does your animal fly? --> ");

n.left = new Node("Does your animal swim? --> ");

n.left.left = new Node("Does your animal have four legs? --> ");

n.left.right = new Node("Does your animal have legs and arms? -->");

n.left.left.left = new Node("Kangaroo");

n.left.left.right = new Node("Dog");

n.left.right.left = new Node("Shark");

n.left.right.right = new Node("Frog");

n.right = new Node("Does it talk? -->");

n.right.left = new Node("Eagle");

n.right.right = new Node("Parrot");

}

Node head = n;

while (true) {

System.out

.print("Welcome to Animal Guess. Think of an animal and type 'y' to start! --> ");

if (sc.nextLine().charAt(0) != 'y') {

System.out

.println("It seems you did not want to play. Goodbye!");

System.exit(0);

}

while (n.left != null) {

System.out.print(n.question);

if (sc.nextLine().charAt(0) == 'y') {

n = n.right;

} else {

n = n.left;

}

}

System.out.print("Is it a " + n.question + "? --> ");

if (sc.nextLine().charAt(0) == 'y') {

System.out.println("Yay! Thanks for playing!");

break;

} else {

System.out.println("Aww! Help me learn!");

System.out.print("What is the name of your animal? --> ");

String animal = sc.nextLine();

System.out

.print("What question would distinguish your animal from "

+ n.question + "? -->");

String question = sc.nextLine();

System.out.print("And would the answer be yes or no? -->");

if (sc.nextLine().charAt(0) == 'y') {

n.right = new Node(animal);

n.left = new Node(n.question);

} else {

n.left = new Node(animal);

n.right = new Node(n.question);

}

n.question = question;

System.out.println("Thank you for teaching me!");

break;

}

}

try {

FileOutputStream ous = new FileOutputStream("tree.ser");

ObjectOutputStream oos = new ObjectOutputStream(ous);

oos.writeObject(head);

oos.close();

} catch (Exception e) {

System.out.println("Could not save file.");

e.printStackTrace();

}

}

}

**Output**

