**Background**

This program plays a animal guessing game in which the user is asked a series of yes/no questions about an animal. Eventually, the program makes a guess as to what the animal is. If the program's response is wrong, the user is asked to supply a yes-no question that can distinguish between the program's guess and the correct answer, and this information is added to the program's knowledge base.

The program's internal knowledge base is represented as a binary decision tree, in which internal nodes represent questions and external nodes represent guesses. When the program guesses wrong, the external node containing the guess is replaced by a new sub tree containing the user-supplied distinguishing question, the original guess, and the new entity.

**Methodology:**

The program was written in java and was implemented using tree data structure.Two classes were made for this Lab assignment; the first one being the source code for animal guessing game and the other one is for the Binary Node. It consists of the following:

1. The actual question.
2. The right answer of question or another list of three elements that signifies the question to be asked if the user responded as ‘yes’ for the earlier question.
3. The Wrong answer of the question or another list of three elements that signifies the question to be asked if the user responded as ‘no for the earlier question

Example:

Does it fly? [Pigeon, butterfly]

Does it live underwater? [Fish]

If user gives a yes to this question then it is assumed that the answer could be either a pigeon or a butterfly. If no then, it is assumed to be a fish or dog and further questions are asked. At the end, the user needs to provide a rhetorical question if that animal is not the one that the user have been thinking about.

**Codes:**

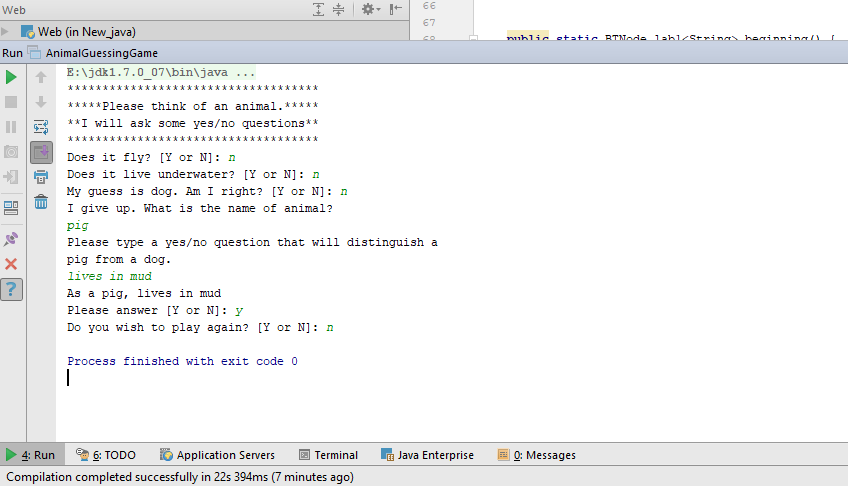
**Source code for animal guessing game:**

**package** com.dwit.bean;  
  
**import** java.util.Scanner;  
  
*/\*\*  
 \* Created by Dell on 2/22/2018.  
 \*/***public class** AnimalGuessingGame {  
 **private static** Scanner *sc* = **new** Scanner(System.***in***);  
  
  
 **public static void** main(String[] args) { */\*The main method prints instructions and repeatedly plays the  
 \* animal-guessing game. As the game is played, the taxonomy tree  
 \* grows by learning new animals.\*/* BTNode\_lab1<String> root;  
  
 *question*();  
 root = *beginning*();  
 **do** {  
 *play*(root);  
 } **while** (*query*(**"Do you wish to play again?"**));  
  
 }  
  
 **public static boolean** query(String prompt) { */\*function to get the valid output from user if user enters invalid  
 answer\*/* String answer;  
  
 System.***out***.print(prompt + **" [Y or N]: "**);  
 answer = *sc*.nextLine().toUpperCase();  
 **while** (!answer.startsWith(**"Y"**) && !answer.startsWith(**"N"**)) {  
 System.***out***.print(**"Invalid response. Please type Y or N: "**);  
 answer = *sc*.nextLine().toUpperCase();  
 }  
  
 **return** answer.startsWith(**"Y"**);  
 }  
  
  
 **public static void** question() {  
 System.***out***.println(**"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"**);  
 System.***out***.println(**"\*\*\*\*\*Please think of an animal.\*\*\*\*\*"**);  
 System.***out***.println(**"\*\*I will ask some yes/no questions\*\*"**);  
  
 System.***out***.println(**"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"**);  
 }  
  
  
 **public static void** play(BTNode\_lab1<String> current) {  
 **while** (!current.isGuess()) {  
 **if** (*query*(current.getData()))  
 current = current.getLeft();  
 **else** current = current.getRight();  
 }  
  
 System.***out***.print(**"My guess is "** + current.getData() + **". "**);  
 **if** (!*query*(**"Am I right?"**))  
 *learn*(current);  
 **else** {  
 System.***out***.println(**"\*\*\*\*\*\*\*\*\*\*\*"**);  
 System.***out***.println(**"\* I won! \*"**);  
 System.***out***.println(**"\*\*\*\*\*\*\*\*\*\*\*"**);  
 }  
 }  
  
  
 **public static** BTNode\_lab1<String> beginning() { *//Construct a small taxonomy tree with four animals.* BTNode\_lab1<String> root;  
 BTNode\_lab1<String> child;  
  
 **final** String ROOT\_QUESTION = **"Does it fly?"**;  
 **final** String LEFT\_QUESTION = **"is it bird?"**;  
 **final** String RIGHT\_QUESTION = **"Does it live underwater?"**;  
 **final** String ANIMAL1 = **"pigeon"**;  
 **final** String ANIMAL2 = **"butterfly"**;  
 **final** String ANIMAL3 = **"fish"**;  
 **final** String ANIMAL4 = **"dog"**;  
  
 *// Create the root node with the question* root = **new** BTNode\_lab1<String>(ROOT\_QUESTION, **null**, **null**);  
  
 *// Create and attach the left subtree.* child = **new** BTNode\_lab1<String>(LEFT\_QUESTION, **null**, **null**);  
 child.setLeft(**new** BTNode\_lab1<String>(ANIMAL1, **null**, **null**));  
 child.setRight(**new** BTNode\_lab1<String>(ANIMAL2, **null**, **null**));  
 root.setLeft(child);  
  
 *// Create and attach the right subtree.* child = **new** BTNode\_lab1<String>(RIGHT\_QUESTION, **null**, **null**);  
 child.setLeft(**new** BTNode\_lab1<String>(ANIMAL3, **null**, **null**));  
 child.setRight(**new** BTNode\_lab1<String>(ANIMAL4, **null**, **null**));  
 root.setRight(child);  
  
 **return** root;  
 }  
  
  
 **public static void** learn(BTNode\_lab1<String> current)  
  
 {  
 String guessAnimal;  
 String correctAnimal;  
 String newQuestion;  
  
 guessAnimal = current.getData();  
 System.***out***.println(**"I give up. What is the name of animal? "**);  
 correctAnimal = *sc*.nextLine();  
 System.***out***.println(**"Please type a yes/no question that will distinguish a"**);  
 System.***out***.println(correctAnimal + **" from a "** + guessAnimal + **"."**);  
 newQuestion = *sc*.nextLine();  
  
 current.setData(newQuestion);  
 System.***out***.println(**"As a "** + correctAnimal + **", "** + newQuestion);  
 **if** (*query*(**"Please answer"**)) {  
 current.setLeft(**new** BTNode\_lab1<String>(correctAnimal, **null**, **null**));  
 current.setRight(**new** BTNode\_lab1<String>(guessAnimal, **null**, **null**));  
 } **else** {  
 current.setLeft(**new** BTNode\_lab1<String>(guessAnimal, **null**, **null**));  
 current.setRight(**new** BTNode\_lab1<String>(correctAnimal, **null**, **null**));  
 }  
 }  
  
}

* **Source code for Binary tree node:**

**package** com.dwit.bean;  
  
*/\*\*  
 \* Created by Dell on 2/22/2018.  
 \*/* **public class** BTNode\_lab1<T>  
 **extends** java.lang.Object  
 {  
  
 **private** String **data**;  
 **private** BTNode\_lab1 **left**, **right**;  
  
  
 **public** BTNode\_lab1(String initialData, BTNode\_lab1 initialLeft, BTNode\_lab1 initialRight)  
 {  
 **data** = initialData;  
 **left** = initialLeft;  
 **right** = initialRight;  
 }  
 **public boolean** isGuess( )  
 {  
 **return** (**left** == **null**) && (**right** == **null**);  
 }  
  
  
 **public void** setData(String newData)  
 {  
 **data** = newData;  
 }  
  
 **public** String getData( )  
 {  
 **return data**;  
 }  
  
  
 **public** BTNode\_lab1 getLeft( )  
 {  
 **return left**;  
 }  
  
  
 **public** BTNode\_lab1 getRight( )  
 {  
 **return right**;  
 }  
 **public void** setLeft(BTNode\_lab1 newLeft)  
 {  
 **left** = newLeft;  
 }  
  
  
 **public void** setRight(BTNode\_lab1 newRight)  
 {  
 **right** = newRight;  
 }  
  
  
  
  
 }

**Output:**



**.Conclusion:**

Hence, Binary tree method was implemented for making the animal guessing game.