```java
package Testing;

import FileParser.FileToList;
import List.LNode;
import AVLTree.AVLTree;

/**
 * COSC 2P03 Assignment 3
 *
 * Created by Matt Laidman on October 11, 2014.
 * Student Number 5199807
 *
 * AVLTest class is a test driver for the AVLTree data structure.
 *
 * ************* Operations Performed ****************
 *
 * inOrder()    Initial In Order traversal of the tree.
 * isAVL()      Check if tree is AVL compliant.
 * delete()     Delete keys starting with D - N and d - n, inclusive.
 * inOrder()    Perform another In Order traversal of the tree.
 * isAVL()      Check if tree is still AVL compliant.
 *
 * @author Matt Laidman (5199807)
 * @version 1.0 (October 21, 2014)
 */
public class AVLTest {

    public AVLTest() {

        LNode words = new FileToList("dat/input.txt").data;     // Get list of words from input file
        AVLTree tree = new AVLTree(words);                      // Build tree from list of words
        System.out.println("\nInOrder Traversal:\n");
        tree.inOrder();                                         // Perform In-Order traversal
        System.out.println("\nTree isAVL: "+tree.isAVL());      // Check if tree is AVL
        System.out.println("\nDeleting Words");
        while (words != null) {                                 // Delete each word starting with dD - nN
            if ((words.key.charAt(0) >= 'D' && words.key.charAt(0) <= 'N') ||
                    (words.key.charAt(0) >= 'd' && words.key.charAt(0) <= 'n')) {
                tree.delete(words.key);
            }
            words = words.next;
        }
        System.out.println("\nInOrder Traversal:\n");
        tree.inOrder();                                         // Perform In-Order traversal
        System.out.println("\nTree isAVL: "+tree.isAVL());      // Check if tree is AVL
    }

    public static void main(String[] args) {
        new AVLTest();
    }
}
```