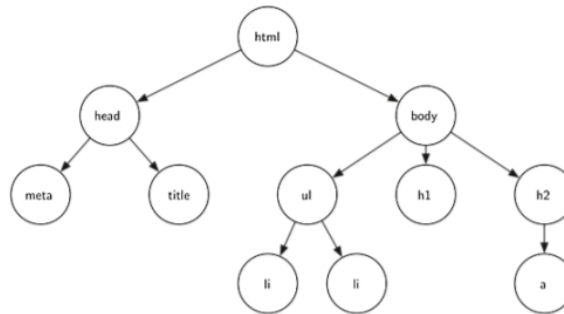


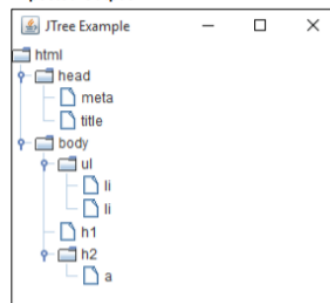
MIDTERMS TASK PERFORMANCE 4

ABLERO, ANGELICA A.

PART 1 (WEEK 8)



Expected Output:



SYNTAX:

```
import javax.swing.*;
import java.awt.*;
import javax.swing.tree.DefaultMutableTreeNode;
import java.util.Collections;

public class HtmlTree {
```

```
    HtmlTree() {
        JFrame f = new JFrame("Tree");
        f.setLayout(new FlowLayout());

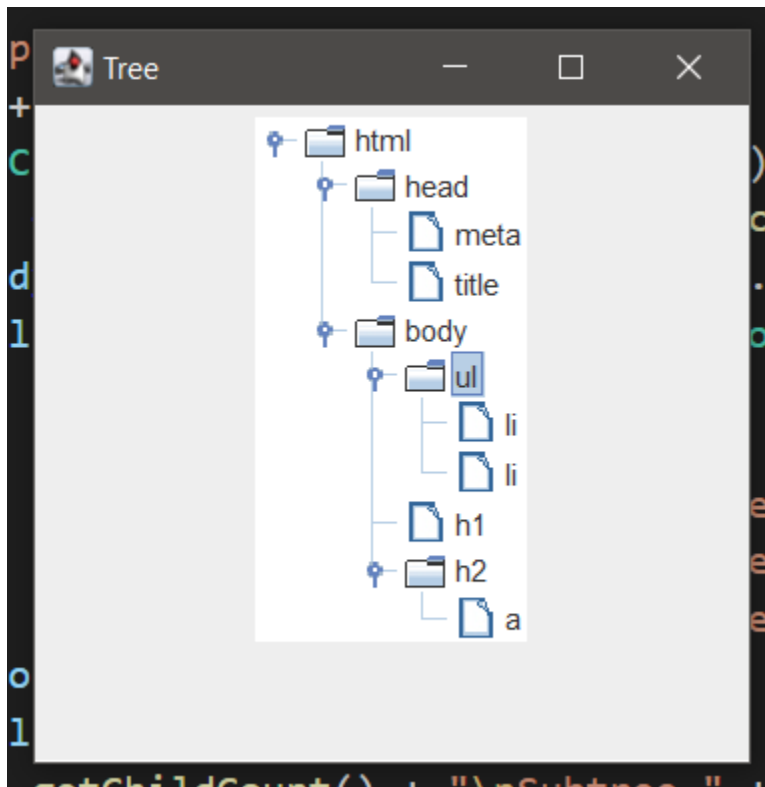
        DefaultMutableTreeNode Html = new DefaultMutableTreeNode("html");
        DefaultMutableTreeNode Head = new DefaultMutableTreeNode("head");
        DefaultMutableTreeNode Meta = new DefaultMutableTreeNode("meta");
        DefaultMutableTreeNode Title = new DefaultMutableTreeNode("title");
        DefaultMutableTreeNode Body = new DefaultMutableTreeNode("body");
        DefaultMutableTreeNode Ul = new DefaultMutableTreeNode("ul");
        DefaultMutableTreeNode Li = new DefaultMutableTreeNode("li");
        DefaultMutableTreeNode Li2 = new DefaultMutableTreeNode("li");
        DefaultMutableTreeNode H1 = new DefaultMutableTreeNode("h1");
        DefaultMutableTreeNode H2 = new DefaultMutableTreeNode("h2");
        DefaultMutableTreeNode A = new DefaultMutableTreeNode("a");
```

```
        JTree tree = new JTree(Html);
        Html.add(Head); Html.add(Body);
        Head.add(Meta); Head.add(Title);
        Body.add(Ul); Body.add(H1); Body.add(H2);
        Ul.add(Li); Ul.add(Li2);
        H2.add(A);

        f.add(tree);
        tree.setShowsRootHandles(true);
        System.out.println(tree.getShowsRootHandles());

        f.setSize(300, 300);
        f.setVisible(true);
        f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

SYNTAX OF PART 1 (WEEK 8)



PART 2 (WEEK 9)

4. Using Java methods and `println()`, enumerate the following:
 - 4.1. Root node
 - 4.2. Parent nodes
 - 4.3. Siblings
 - 4.4. One-level subtrees
 - 4.5. Nodes per level
 - 4.6. Depth
 - 4.7. Degree of each one-level subtree
 - 4.8. List of nodes based on breadth-first, preorder, and postorder

SYNTAX:

```
System.out.println("Root node: " + Html.getRoot() + "\n");

System.out.println("Parent nodes: " + Head.getParent() + ", " +
Meta.getParent() + ", " + Ul.getParent() + ", " + Li.getParent() + ", " + A.getParent() + "\n");

System.out.println("Siblings: \n" + Collections.list(Html.children()) + "\n" +
Collections.list(Head.children()) + "\n" + Collections.list(Body.children()) + "\n" +
Collections.list(Ul.children()) + "\n" + Collections.list(H2.children()) + "\n");

System.out.println("One-level subtrees \n" + Html.getRoot() + " - " + Collections.list(Html.children()) +
"\n" + Meta.getParent() + " - " + Collections.list(Head.children()) + "\n" + H1.getParent() +
" - " + Collections.list(Body.children()) + "\n" + Li.getParent() + " - " + Collections.list(Ul.children())
+ "\n" + A.getParent() + " - " + Collections.list(H2.children())
+ "\n" );

System.out.println("Nodes per level: \n"+ "Level "
+ Html.getLevel() + " - " + Html.getRoot() + "\nLevel " +
Body.getLevel() + " - " + Collections.list(Html.children()) +
"\nLevel " + Ul.getLevel() + " - " + Collections.list(Head.children()) +
" - " + Collections.list(Body.children()) + "\nLevel " + Li.getLevel() +
" - " + Collections.list(Ul.children()) + " - " + Collections.list(H2.children()) + "\n" );

System.out.println("Depth: " + Html.getDepth() + "\n");
System.out.println("Degree of each one-level \n" + "Subtree " +
Head.getParent() + " - " + Html.getChildCount() + "\nSubtree " +
Meta.getParent() + " - " + Head.getChildCount() + "\nSubtree " +
Ul.getParent() + " - " + Body.getChildCount() + "\nSubtree " +
Li.getParent() + " - " + Ul.getChildCount() + "\nSubtree " +
A.getParent() + " - " + H2.getChildCount() + "\nSubtree " + "\n");
```

```
System.out.println("List of nodes based on breadth-first, preorder, and postorder \n" + "\nBreadth-first: " +
Collections.list(Html.breadthFirstEnumeration()) + "\nPreorder: " + Collections.list(Html.preorderEnumeration())
+ "\nPosorder: " + Collections.list(Html.postorderEnumeration()));
```

OUTPUT:

```
PS C:\Users\Gie> & 'C:\Program Files\Java\jdk-11\bin\java.exe' '-cp' 'C:\Users\Gie\AppData\Local\Temp\vscodesws_6559d\jdt_ws\jdt.ls-java-project\bin' 'HtmlTr
ee'
true
Root node: html
```

```
Parent nodes: html, head, body, ul, h2
```

```
Siblings:
[head, body]
[meta, title]
[ul, h1, h2]
[li, li]
[a]
```

```
One-level subtrees
```

```
html - [head, body]
head - [meta, title]
body - [ul, h1, h2]
ul - [li, li]
h2 - [a]
```

```
Nodes per level:
```

```
Level 0 - html
Level 1 - [head, body]
Level 2 - [meta, title], [ul, h1, h2]
Level 3 - [li, li] - [a]
```

```
Depth: 3
```

```
Degree of each one-level
```

```
Subtree html - 2
Subtree head - 2
Subtree body - 3
Subtree ul - 2
Subtree h2 - 1
```

```
List of nodes based on breadth-first, preorder, and postorder
```

```
Breadth-first: [html, head, body, meta, title, ul, h1, h2, li, li, a]
Preorder: [html, head, meta, title, body, ul, li, li, h1, h2, a]
Posorder: [meta, title, head, li, li, ul, h1, a, h2, body, html]
```

WHOLE SYNTAX:

```
import javax.swing.*;
import java.awt.*;
import javax.swing.tree.DefaultMutableTreeNode;
import java.util.Collections;
public class HtmlTree {

    HtmlTree(){
        JFrame f = new JFrame("Tree");
        f.setLayout(new FlowLayout());

        DefaultMutableTreeNode Html = new DefaultMutableTreeNode("html");
        DefaultMutableTreeNode Head = new DefaultMutableTreeNode("head");
        DefaultMutableTreeNode Meta = new DefaultMutableTreeNode("meta");
        DefaultMutableTreeNode Title = new DefaultMutableTreeNode("title");
        DefaultMutableTreeNode Body = new DefaultMutableTreeNode("body");
        DefaultMutableTreeNode U1 = new DefaultMutableTreeNode("ul");
        DefaultMutableTreeNode Li = new DefaultMutableTreeNode("li");
        DefaultMutableTreeNode Li2 = new DefaultMutableTreeNode("li");
        DefaultMutableTreeNode H1 = new DefaultMutableTreeNode("h1");
        DefaultMutableTreeNode H2 = new DefaultMutableTreeNode("h2");
        DefaultMutableTreeNode A = new DefaultMutableTreeNode("a");

        JTree tree = new JTree(Html);
        Html.add(Head); Html.add(Body);
        Head.add(Meta); Head.add(Title);
        Body.add(U1); Body.add(H1); Body.add(H2);
        U1.add(Li); U1.add(Li2);
        H2.add(A);

        f.add(tree);
        tree.setShowsRootHandles(true);
        System.out.println(tree.getShowsRootHandles());

        f.setSize(300, 300);
        f.setVisible(true);
        f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        System.out.println("Root node: " + Html.getRoot() + "\n");

        System.out.println("Parent nodes: " + Head.getParent() + ", " +
            Meta.getParent() + ", " + U1.getParent() + ", " + Li.getParent() + ", " + A.getParent() + "\n");

        System.out.println("Siblings: \n" + Collections.list(Html.children()) + "\n" +
            Collections.list(Head.children()) + "\n" + Collections.list(Body.children()) + "\n" +
            Collections.list(U1.children()) + "\n" + Collections.list(H2.children()) + "\n");

        System.out.println("One-level subtrees \n" + Html.getRoot() + " - " + Collections.list(Html.children()) +
            "\n" + Meta.getParent() + " - " + Collections.list(Head.children()) + "\n" + H1.getParent() +
            " - " + Collections.list(Body.children()) + "\n" + Li.getParent() + " - " + Collections.list(U1.children())
            + "\n" + A.getParent() + " - " + Collections.list(H2.children())
            + "\n" );

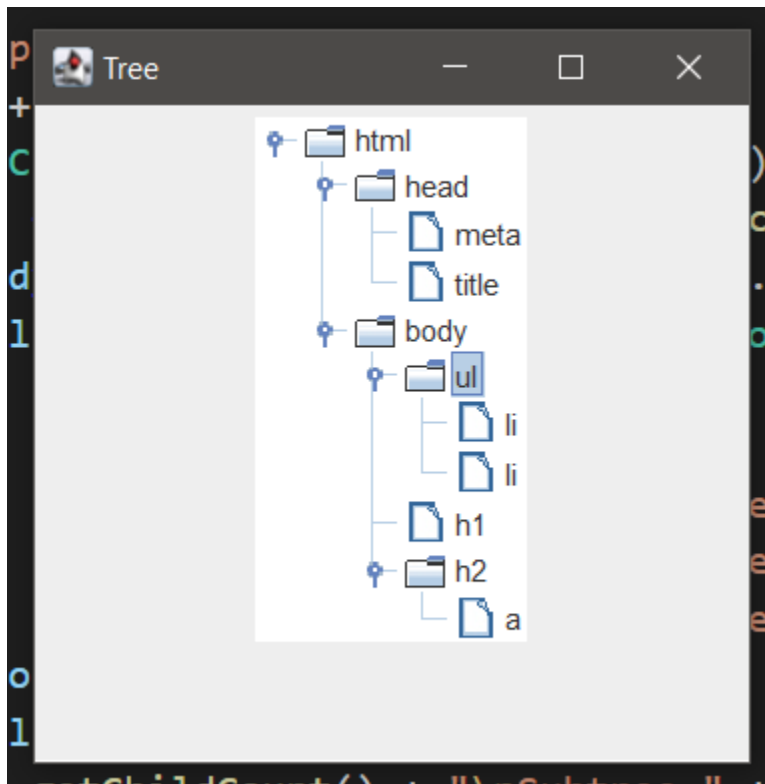
        System.out.println("Nodes per level: \n"+ "Level "
            + Html.getLevel() + " - " + Html.getRoot() + "\nLevel " +
            Body.getLevel() + " - " + Collections.list(Html.children()) +
            "\nLevel " + U1.getLevel() + " - " + Collections.list(Head.children()) +
            ", " + Collections.list(Body.children()) + "\nLevel " + Li.getLevel() +
            " - " + Collections.list(U1.children()) + " - " + Collections.list(H2.children()) + "\n" );

        System.out.println("Depth: " + Html.getDepth() + "\n");
        System.out.println("Degree of each one-level \n" + "Subtree " +
            Head.getParent() + " - " + Html.getChildCount() + "\nSubtree " +
            Meta.getParent() + " - " + Head.getChildCount() + "\nSubtree " +
            U1.getParent() + " - " + Body.getChildCount() + "\nSubtree " +
            Li.getParent() + " - " + U1.getChildCount() + "\nSubtree " +
            A.getParent() + " - " + H2.getChildCount() + "\n");

        System.out.println("List of nodes based on breadth-first, preorder, and postorder \n" + "\nBreadth-first: " +
            Collections.list(Html.breadthFirstEnumeration()) + "\nPreorder: " + Collections.list(Html.preorderEnumeration())
            + "\nPosorder: " + Collections.list(Html.postorderEnumeration()));
    }

    Run | Debug
    public static void main(String[] args) {
        HtmlTree tree = new HtmlTree();
        System.out.println(tree);
    }
}
```

OUTPUT:



```
PS C:\Users\Gie> & 'C:\Program Files\Java\jdk-11\bin\java.exe' '-cp' 'C:\Users\Gie\AppData\Local\Temp\vscodesws_6559d\jdt_ws\jdt.ls-java-project\bin' 'HtmlTree'
true
Root node: html
```

```
Parent nodes: html, head, body, ul, h2
```

```
Siblings:
[head, body]
[meta, title]
[ul, h1, h2]
[li, li]
[a]
```

```
One-level subtrees
html - [head, body]
head - [meta, title]
body - [ul, h1, h2]
ul - [li, li]
h2 - [a]
```

```
Nodes per level:
Level 0 - html
Level 1 - [head, body]
Level 2 - [meta, title], [ul, h1, h2]
Level 3 - [li, li] - [a]
```

```
Depth: 3
```

```
Degree of each one-level
Subtree html - 2
Subtree head - 2
Subtree body - 3
Subtree ul - 2
Subtree h2 - 1
```

```
List of nodes based on breadth-first, preorder, and postorder
```

```
Breadth-first: [html, head, body, meta, title, ul, h1, h2, li, li, a]
Preorder: [html, head, meta, title, body, ul, li, li, h1, h2, a]
Postorder: [meta, title, head, li, li, ul, h1, a, h2, body, html]
```