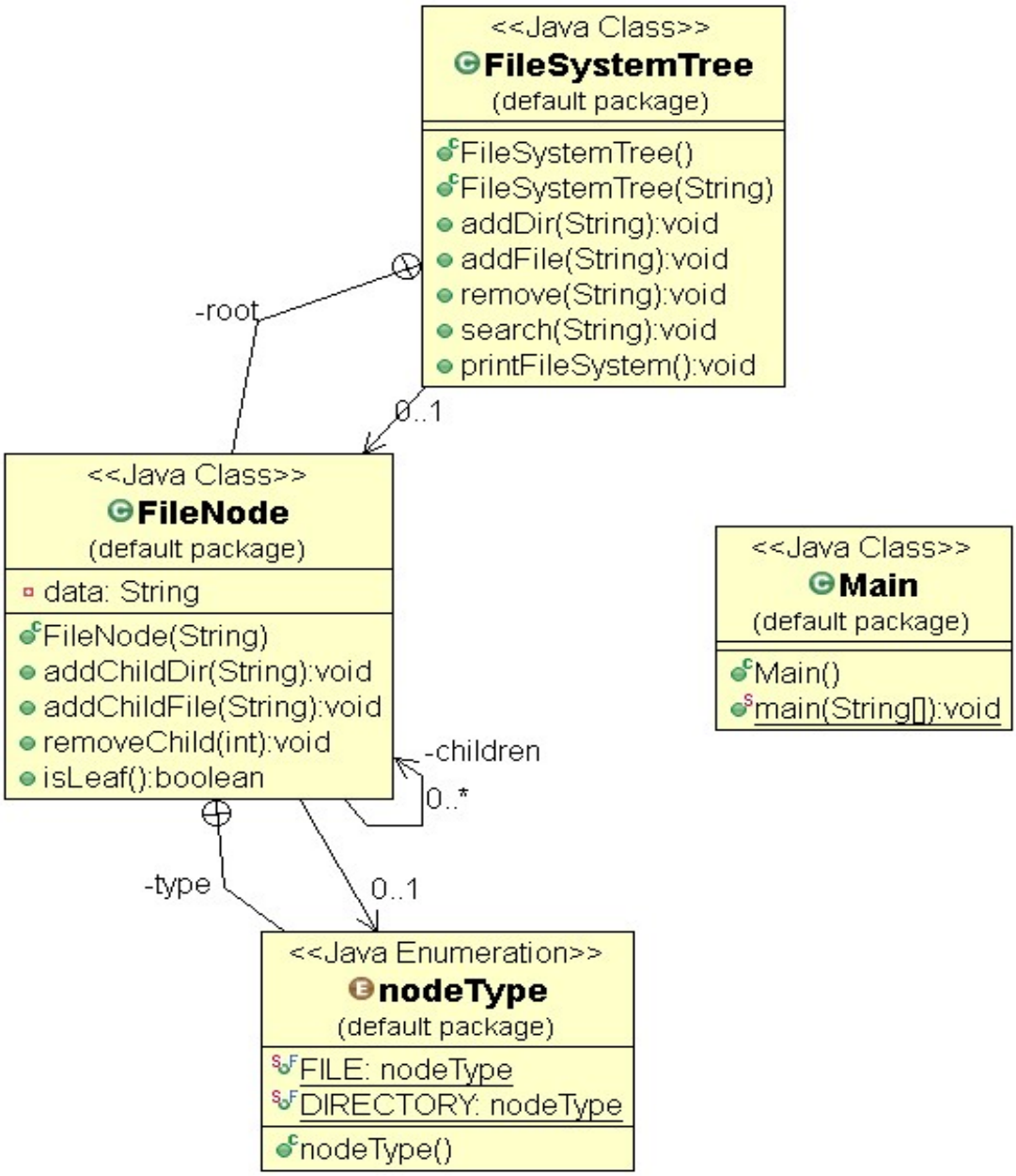
GTU Department of Computer Engineering CSE 222/505 - Spring 2020
Homework 05 Report

ESRA ERYILMAZ
171044046

- Q1 -

1. **CLASS DIAGRAM**



2. PROBLEM SOLUTION APPROACH

Inside this file system I keep FileNode class to handle the nodes. Also inside file system tree, I keep file node root.
In File node class there is enum for deciding node is a file or directory.
In add methods I add directory or file to the file system. While adding there is a queue for traversing the tree. If temporary node is in the right node than it adds.
In remove method , it removes file or directory. If given directory has child ; it asks user whether remove or not. And also if given path cannot found it warns user.

While adding removing and printing I use level order traverse.
In search I want to use preorder traversal but I could not implement it.

3. TEST CASES

Test Case ID	Test Method	Test Input	Test Output	Pass/Fail
T1	addDir() method	Directory path (as string)	Void	Pass
T2	addFile() method	File path (as string)	Void	Pass
T3	remove() method	Directory or file path (as string)	Void	Pass
T4	search() method			Fail
T5	printFileSystem() method	No input	Prints the system path as string	Pass

4. RUNNING AND RESULTS

Test ID	Test Results

T1	`> addDir("root/first_directory")`
	`[PRINT]`
	`root`
	`first_directory`

T2	`> addFile("root/first_directory/new_file.txt")`
	`[PRINT]`
	`root`
	`first_directory second_directory`
	`new_file.txt`
	`> addFile("root/second_directory/new_directory/new_file.doc")`
	`[PRINT]`
	`root`
	`first_directory second_directory`
	`new_file.txt new_directory`
	`new_file.doc`

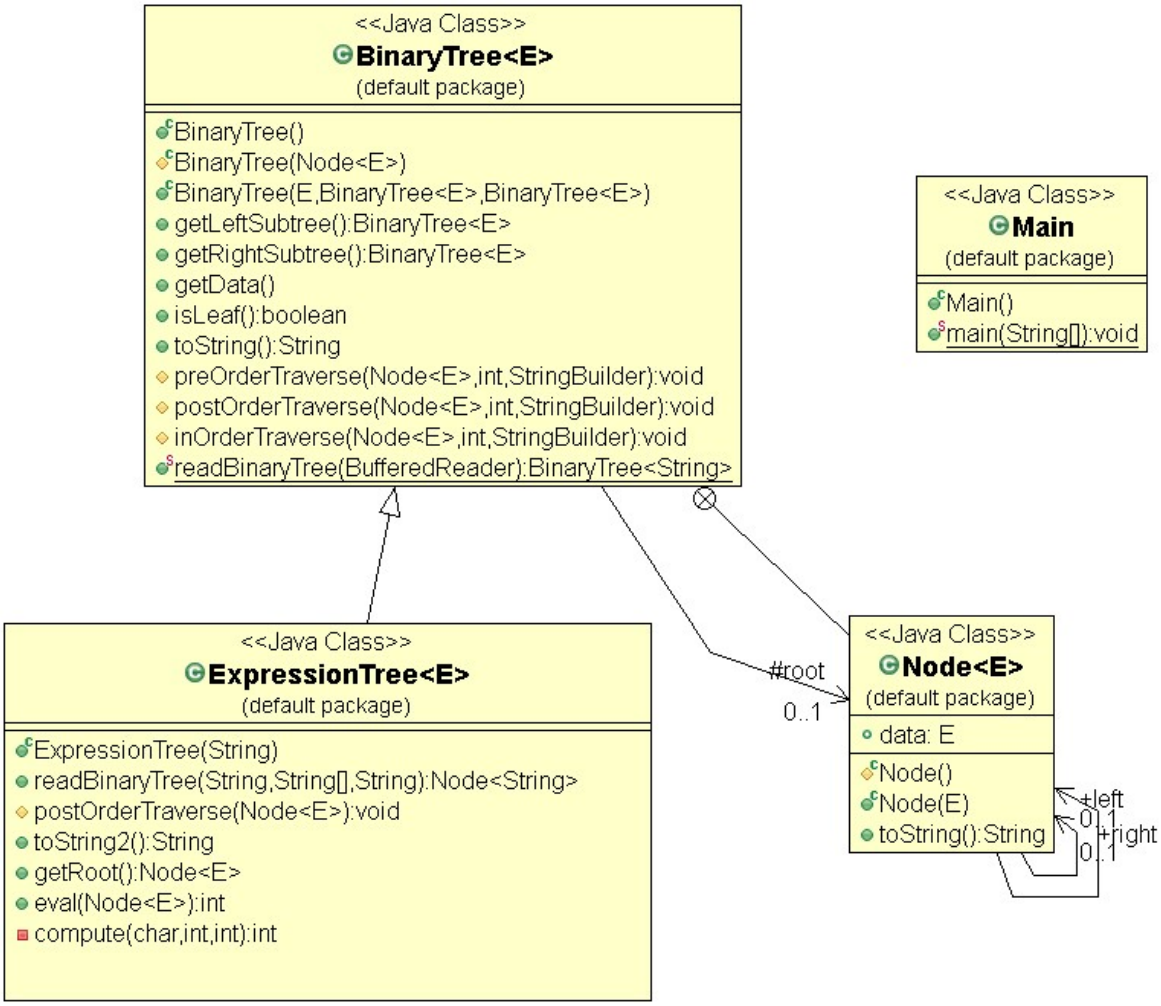
T3	`> remove("root/first_directory/new_file.txt")`
	`[PRINT]`
	`root`
	`first_directory second_directory`
	`new_directory`
	`new_file.doc`
	
	`> remove("root/second_directory/new_directory")`
	`Directory has child. Do you want to delete it?`
	` YES>1`
	` NO>0`
	`1`
	`[PRINT]`
	`root`
	`first_directory second_directory`

T4	-

T5	`[PRINT]`
	`root`
	`first_directory second_directory`
	`new_file.txt new_directory`
	`new_file.doc`

- Q2 -

1. CLASS DIAGRAMS



2. PROBLEM SOLUTION APPROACH

While creating binary tree I use readBinaryTree(). But since it is static could not override it. I create my own readBinaryTree.

I construct tree recursively. But sometimes I don't know why, it does not take last element in the expression.

My iteration works like; it takes element and according to the element, if element is numeric it returns node ,if element is operand it creates tree and iteration works. It creates operand's left and right child.

PostOrderTraverse() method : It prints nodes in post order traversal shape. Using iteration.

toString2() method : It just calls PostOrderTraverse().

Eval() method : It evaluates expression and returns result;

3. TEST CASES

Test Case ID	Test Method	Test Input	Test Output	Pass/Fail
T1	ExpressionTree() method (Constructor)	Expression (as String)	It calls readBinaryTree()	Pass
T2	readBinaryTree() method	Expressions (as String and String array)	It creates binary tree (But for some reason I don't know why, sometimes it doesn't take the last element in the expression)	Middle pass 😊
T3	postOrderTraverse() method	Root node	It prints nodes	Pass
T4	toString2() method	No input	It calls postOrderTraverse()	Pass
T5	eval() method	Root node	Returns expression result.	Pass

P.S.

Other functions may appear like it doesn't work right , because the constructor (or readBinaryTree) does not always construct binary tree correctly.

In other words ,other methods may have used the wrongly created tree when processing.

4. RUNNING AND RESULTS

Test ID	Test Results
T1	Constructor creates tree by calling readBinaryTree().
T2	Creates tree.
T3	Prints tree post order traverse
T4	Calls postOrderTraverse()S toString2() = 10 5 15 * + 10 + toString2() = 15 5 15 * + 20 +
T5	Evlauates expression and return it.

Running results

```
                TEST STARTING...
---------------------------------------
        [PREFIX TEST]

Prefix exp =  + + 10 * 5 15 20
eval() =  95
toString2() =  10 5 15 * + 10 +

Prefix exp =  + 9 * 2 6
eval() =  21
toString2() =  9 2 6 * +

---------------------------------------

        [POSTFIX]

Postfix exp =  10 5 15 * + 20 +
eval() =  110
toString2() =  15 5 15 * + 20 +

Postfix exp =  9 2 6 * +
eval() =  18
toString2() =  6 2 6 * +
---------------------------------------
                TEST FINISHED...
```

- Q3 -

I could not implement it.

- Q4 -

I could not implement it.