GTU Department of Computer Engineering

CSE 222/505 - Spring 2020

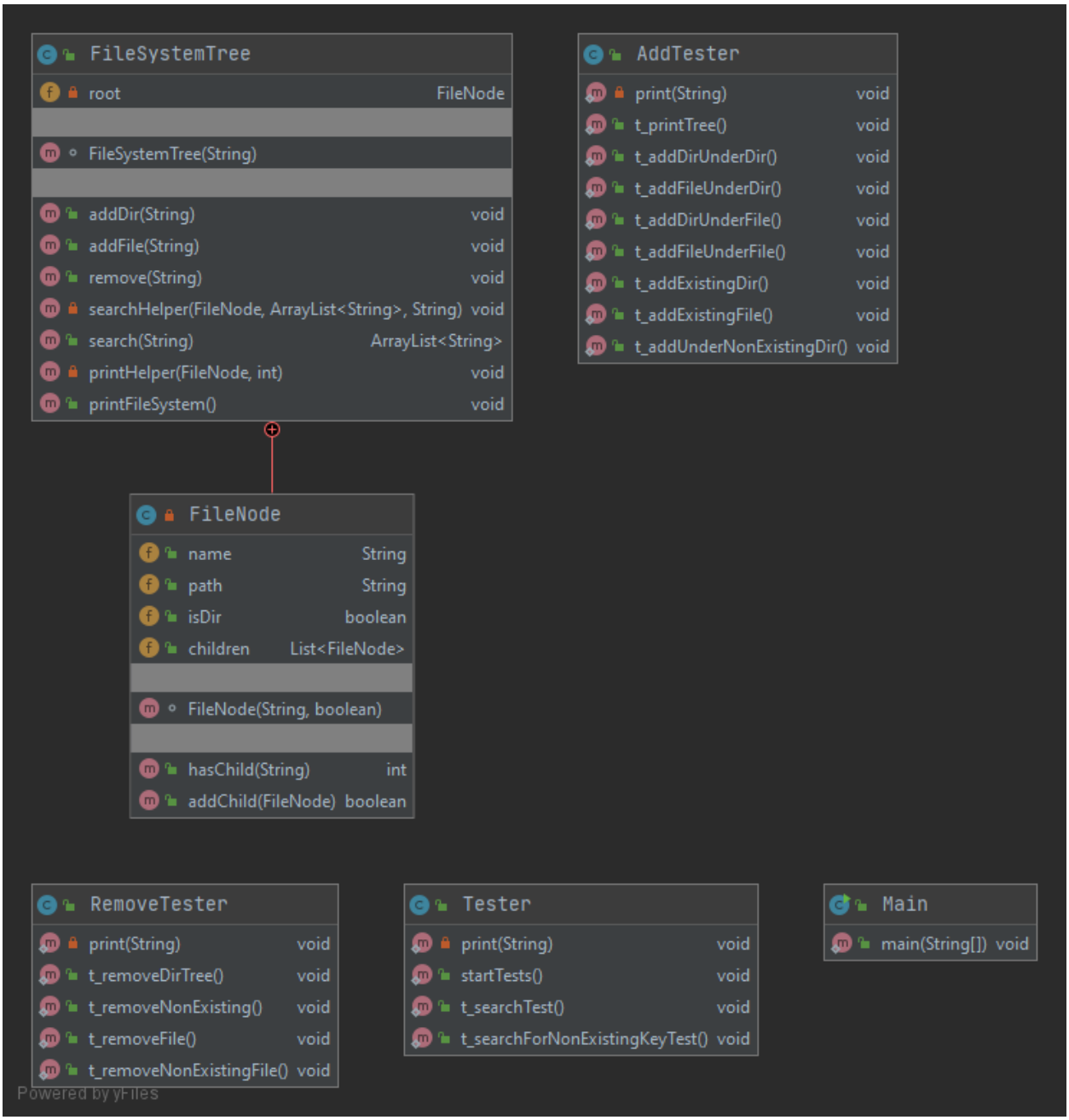
Homework 4 Report

Q1

Buğra Eren Yılmaz

1801042669

1. Class Diagrams



2.Problem Solution approach

The problem at hand was to create a representation of real life computer file system tree that is also functional.

First of I created the FileNode class which would be a node of the future FileSystemTree. It solves the problem of distinguishing files from directories by just defining a simple Boolean flag. Ever node may have any number of children so at this level I abstracted the holding children of nodes by implementing this logic inside FileNode class.

After that, I created the FileSystemTree class, which is a general tree structure that uses FileNode as the tree nodes.

Next problem was the path string tokenization. The inputs will be in full path formats so I needed a way to tokenize and validate the path strings.

After doing the tokenization and ordinary checks, the functions can continue in normal order.

For adding a file or folder to tree, basic traversing of general tree used and functioned. On any kind of user-related error, the class throws FileSystemException by informing the user about error.

The related errors are mostly non-valid system paths.

All of the methods are implemented recursively since this was the most practical way for working with a general tree.

3.Test Cases

Test ID	Test Scenario	Test Steps	Test Data	Expected Results	Actual Results	Pass/Fail
T01	Construct the file system given on pdf and print it	Call the addDir and addFile methods in order.	Empty File System Tree	File System tree given on the pdf	As expected	pass
T02	Add a directory under another directory	Construct initial tree, then add directory	Initial tree with one directory	New directory created under initial tree	As expected	pass
T03	Add a file under directory	Construct initial tree, then add file	Initial tree with one directory	New file created under initial tree	As expected	pass
T04	Add a directory under a file	Construct initial tree, then add directory under file	Initial tree with one directory and file	Exception thrown	As expected	pass
T05	Add a file under a file	Construct initial tree, then add file under file	Initial tree with one directory and file	Exception thrown	As expected	pass
T06	Add a directory that	Construct initial tree, then add	Initial tree with one	Exception thrown	As expected	pass

	already exists	existing directory	directory and file			
T07	Add a file that already exists	Construct initial tree, then add existing file	Initial tree with one directory and file	Exception thrown	As expected	pass
T08	Add under non existing directory	Construct initial tree, then add under non existing directory	Initial tree with one directory and file	Exception thrown	As expected	pass
T09	Remove directory that contains multiple things in it	Construct initial tree, then remove directory	Initial tree with multiple directory and file	User will see the contents of the target and then prompted with yes or no	As expected	pass
T10	Remove non existing directory	Construct initial tree, then remove non existing directory	Initial tree with multiple directory and file	Exception thrown	As expected	pass
T11	Remove file	Construct initial tree, then	Initial tree with multiple	File is removed	As expected	pass

		remove file	directory and file			
T12	Remove non existing file	Construct initial tree, then remove non existing file	Initial tree with multiple directory and file	Throws exception	As expected	pass
T13	Search example given in pdf	Construct initial tree, then search for keyword	Initial tree with multiple directory and file	Returns arraylist of strings that contains the keyword	As expected	pass
T14	Search for non - existing keyword	Construct initial tree, then search for keyword	Initial tree with multiple directory and file	Returns empty arratlist	As expected	pass

4.Running results

Test: Adding some directories and files to demonstrate printing of a tree.

root

----first_directory

----new_file.txt

----second_directory

----new_directory

----new_file.doc

Test: Adding a directory under another directory

Initial tree:

root

----first_directory

After adding directory under first_directory:

root

----first_directory

----second_directory

Test: Adding a file under another directory

Initial tree:

root

----first_directory

After adding file Foo.txt under first_directory:

root

----first_directory

----Foo.txt

Test: Trying to add a directory under a file

Initial tree:

root

----first_directory

----Foo.txt

Trying to add second_directory under Foo.txt:

java.nio.file.FileSystemException: root/first_directory/Foo.txt/second_directory

root

----first_directory

----Foo.txt

Test: Trying to add a file under another file

Initial tree:

root

----first_directory

----Foo.txt

Trying to add file homework.txt under Foo.txt:

java.nio.file.FileSystemException: root/first_directory/Foo.txt/homework.txt

root

----first_directory

----Foo.txt

Test: Trying to add a directory that already exists

Initial tree:

root

----first_directory

----hw.txt

----second_directory

Trying to add second_directory again:

java.nio.file.FileSystemException: root/second_directory

root

----first_directory

----hw.txt

----second_directory

Test: Trying to add a file that already exists

Initial tree:

root

----first_directory

----hw.txt

----second_directory

Trying to add hw.txt under first_directory, again:

java.nio.file.FileSystemException: root/first_directory/hw.txt

root

----first_directory

----hw.txt

----second_directory

Test: Trying to add a file under non existing directory

Initial tree:

root

----first_directory

----hw.txt

----second_directory

Trying to add foo.txt under root/second/third/fourth:

java.nio.file.FileSystemException: root/second/third/fourth/foo.txt

root

----first_directory

----hw.txt

----second_directory

Test: Remove a dir that contains a lot of things

Initial tree:

root

----first_directory

----new_file.txt

----second_directory

----new_directory

----new_file.doc

After request removing second_directory, it will print the contents of target and prompt user with y or n:

second_directory is not empty!

Contents of second_directory:

second_directory

----new_directory

----new_file.doc

Delete anyway? (y/n): y

second_directory deleted!

root

----first_directory

----new_file.txt

Test: Remove a non existing dir

Initial tree:

root

----first_directory

----new_file.txt

----second_directory

----new_directory

----new_file.doc

Trying to remove root/secdir:

java.nio.file.FileSystemException: root/secdir

root

----first_directory

----new_file.txt

----second_directory

----new_directory

----new_file.doc

Test: Remove a file

Initial tree:

root

----first_directory

----new_file.txt

----second_directory

----new_directory

----new_file.doc

Trying to remove root/first_directory/new_file.txt:

root

----first_directory

----second_directory

----new_directory

----new_file.doc

Test: Remove a non existing file

Initial tree:

root

----first_directory

----new_file.txt

----second_directory

----new_directory

----new_file.doc

Trying to remove root/foo.txt:

java.nio.file.FileSystemException: root/foo.txt

root

----first_directory

----new_file.txt

----second_directory

----new_directory

----new_file.doc

Test: Searching a keyword in filesystem

Initial tree:

root

----first_directory

----new_file.txt

----second_directory

----new_directory

----new_file.doc

Searching for keyword 'new', it will return a list of path strings:

file - root/first_directory/new_file.txt

dir - root/second_directory/new_directory

file - root/second_directory/new_directory/new_file.doc

Test: Searching a keyword in filesystem, that does not exist

Initial tree:

root

----first_directory

----new_file.txt

----second_directory

----new_directory

----new_file.doc

Searching for keyword 'CSE', it will return a empty list: