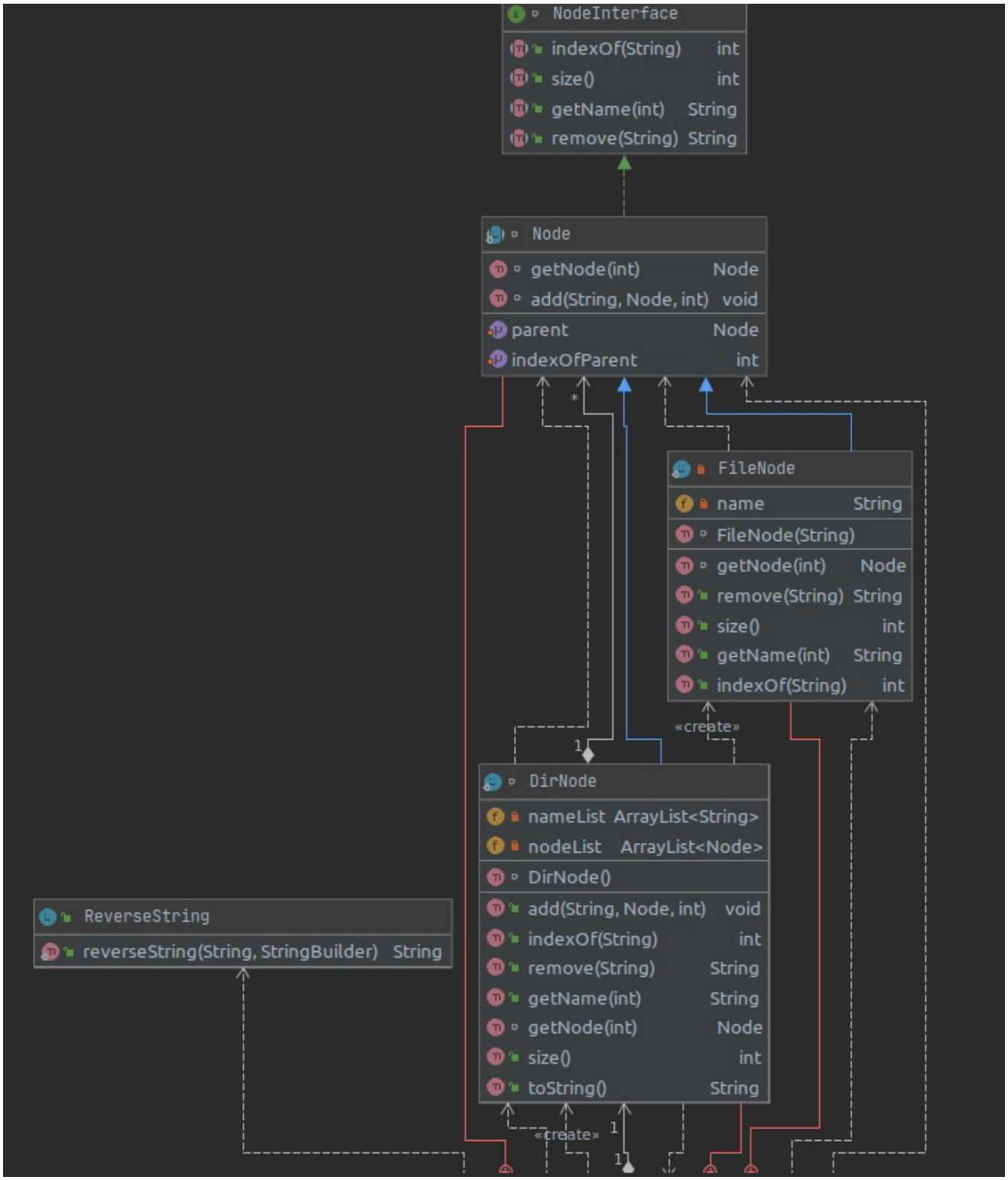
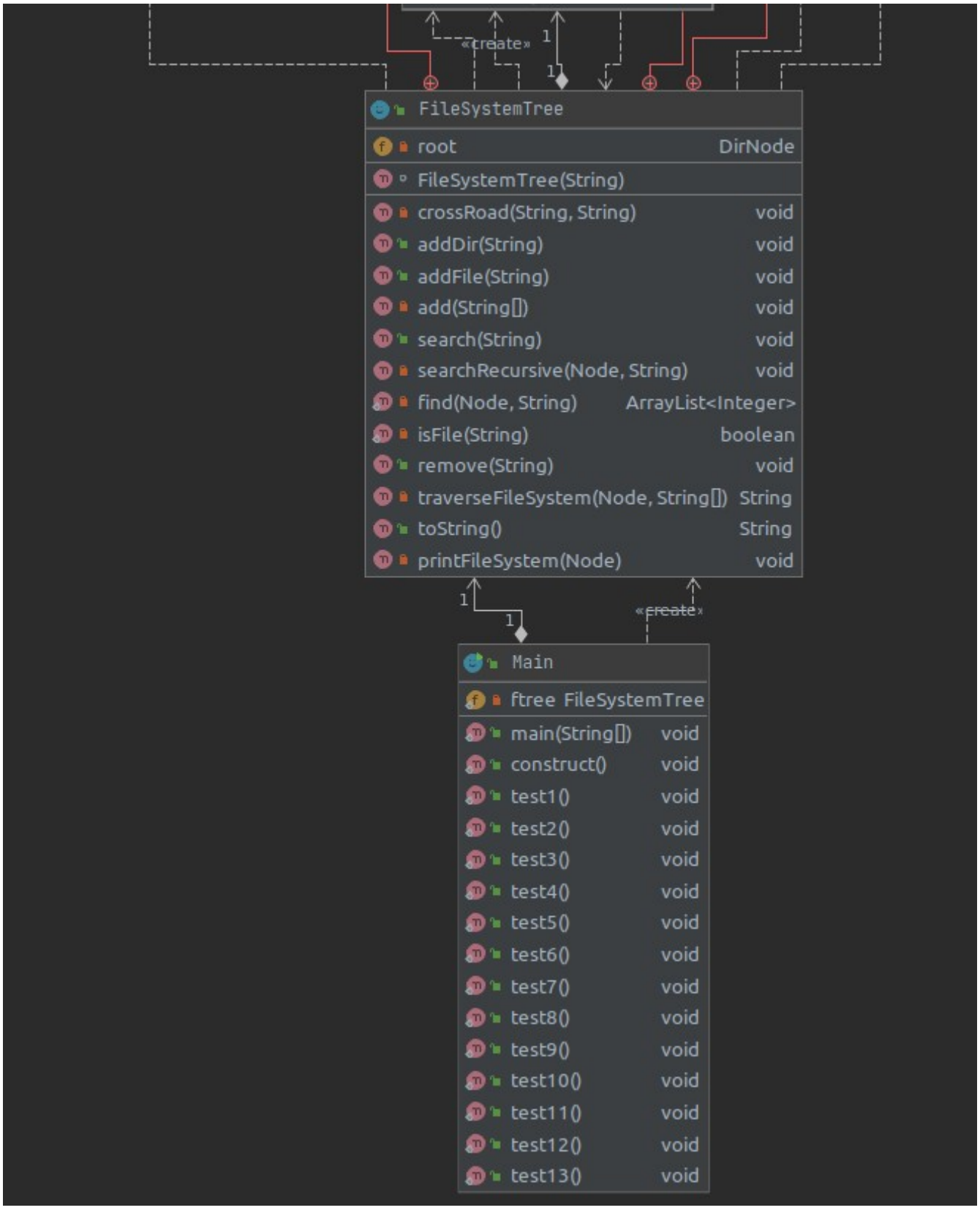
GIT Department of Computer Engineering
CSE 222/505 - Spring 2020
Homework 5 Question 1 Report

Melihcan Çilek
1801042092

1-)Class Diagram




«create»
1
1
FileSystemTree
root DirNode
FileSystemTree(String)
crossRoad(String, String) void
addDir(String) void
addFile(String) void
add(String[]) void
search(String) void
searchRecursive(Node, String) void
find(Node, String) ArrayList<Integer>
isFile(String) boolean
remove(String) void
traverseFileSystem(Node, String[]) String
toString() String
printFileSystem(Node) void
1
1
«create»
Main
ftree FileSystemTree
main(String[]) void
construct() void
test1() void
test2() void
test3() void
test4() void
test5() void
test6() void
test7() void
test8() void
test9() void
test10() void
test11() void
test12() void
test13() void

2-)Problem Solution Approach

If I have to summarize problem for questions 1, problem for question 1 is that implement FileSystemTree class to handle a file system hierarchy in a general tree structure. For this problem, first thing that I recognize is that when we generate a data structure that has nodes, in these nodes we have generic classes such that Integer, Double or custom classses but in this problem, we can have more than one data in a node so we have to use List data structure to hold directories and files as datas. As I recognize this problem, I recognize another thing which is important in my opinion because we will hold either directory or file and these both have some similarities so we have to use one parent class for holding similarity for these classes and I created Node class as parent class and DirNode and FileNode classes are extends from that class so that I can create List structure as Node parameter so it has to support both FileNode and DirNode. After solve this problem, I started implementation phase. In this phase, problems that I encounter are first, as I add either directory or file, both addition has some similarity as it in classes so I took these similarities to a common method. Second, as I find a name, I was looking for only one name inside a directory so after found that name, I return that name so other names wouldn't be found so it could be a problem. For solve that problem, I canceled return option so it is not returning untill look up all files and directories inside that directory. Third, for solving remove directory and if there is subdirectories, asking user for "Do you want to delete?" selection is not that hard as I expected so I solved that problem with if putting one if statement to loop so if I found directory that will be deleted, I ask user that question for getting permission.

3-)Running Command And Results

```
//Using addDir method without initiliazing root directory
public static void test1(){
    ftree.addDir( path: "root/first_directory");
}
```

```
//Creating file as root
public static void test2(){
    ftree = new FileSystemTree( root: "root.txt");
}
```

```
//Creating file as root and adding directory after it as its child
public static void test3(){
    ftree = new FileSystemTree( root: "root.txt");
    ftree.addDir( path: "root.txt/first_directory");
}
```

```
//Creating file as root and adding file after it as its child
public static void test4(){
    ftree = new FileSystemTree( root: "root.txt");
    ftree.addFile( path: "root.txt/first_directory.txt");
}
```

```
//Creating directory as addFile method
public static void test5(){
    ftree = new FileSystemTree( root: "root");
    ftree.addFile( path: "root/first_directory");
}
```

```
//Creating directory as addFile method
public static void test6(){
    ftree = new FileSystemTree( root: "root");
    ftree.addDir( path: "root/first_directory.txt");
}
```

```
//Search method for search file
public static void test7(){
    construct();
    ftree.search( stringWillBeSearched: "home.dart");
}
```

```java
//Search method for search directory
public static void test8(){
    construct();
    ftree.search( stringWillBeSearched: "directory");
}
```

```java
//Remove method for remove empty directory
public static void test9(){
    construct();
    ftree.remove( path: "");
}
```

```java
//Remove method for remove file
public static void test10(){
    construct();
    ftree.remove( path: "root/second_directory/Codeshop/FlutterShop/Larsca/lib/pages/home.dart");
}
```

```java
//Remove method for remove empty directory
public static void test11(){
    construct();
    ftree.remove( path: "root/first_directory");
}
```

```java
//Remove method for remove non-empty directory
public static void test12(){
    construct();
    ftree.remove( path: "root/second_directory/Codeshop");
    System.out.println(ftree);
}
```

```java
//Remove method for remove root
public static void test13(){
    construct();
    ftree.remove( path: "root");
    System.out.println(ftree);
}
```

Note: Some of the results of tests are really long so I did’t took picture of results of tests. I added results of tests into Test Case Table.