Implement **FileSystemTree** class to handle a file system hierarchy in a general tree structure. You need to implement **FileNode** class to handle the nodes of the tree. A node can be created either for a file or a directory. You will decide how discrimination is done between files and directories.

Your **FileSystemTree** implementation must have the following:

A constructor to create a file system with a root directory. Name of the root directory will be given as a parameter to the constructor.

addDir and **addFile** methods to add directories or files to the file system. The path of the new directory (or file) will be given as a parameter to the method.

remove method to remove a directory (or a file) from the file system. The path of the directory (or the file) will be given as a parameter to the method. The method will warn the user if the path cannot be found. If the directory includes some other directories (or files), method will list the contents and ask the user whether to remove or not.

search method to search the entire file system for a directory or a file including the given search characters in its name. The search characters will be given as the parameter of the method.

printFileSystem method to print the whole tree.

Give information about your traversal methods in your javadoc file.

Here are some code lines to help you understand what we expect:

```
//Create a file system with root directory
FileSystemTree myFileSystem = new FileSystemTree("root");


//Add directories and files using paths
myFileSystem.addDir("root/first_directory");
myFileSystem.addDir("root/second_directory");
myFileSystem.addFile("root/first_directory/new_file.txt");
myFileSystem.addDir("root/second_directory/new_directory");
myFileSystem.addFile("root/second_directory/new_directory/new_file
.doc");
```

```


//Search file or directory names including “new”
myFileSystem.search("new");
//This will output:
// file - root/first_directory/new_file.txt
// dir - root/second_directory/new_directory
// file - root/second_directory/new_directory/new_file.doc


//Remove files or directories
myFileSystem.remove("root/first_directory/new_file.txt");
myFileSystem.remove("root/second_directory/new_directory");
```