ACS-3913-001
Assignment 4
Due by Wednesday, December 7 at 11:59 pm

Read and review the GitHub documentation on Nexus. All code, documents, and diagrams will be submitted through GitHub Classroom for this course.

To accept Assignment 1: https://classroom.github.com/a/orylzT3v

- *If you have not done so already, you will first claim your name with your GitHub account*

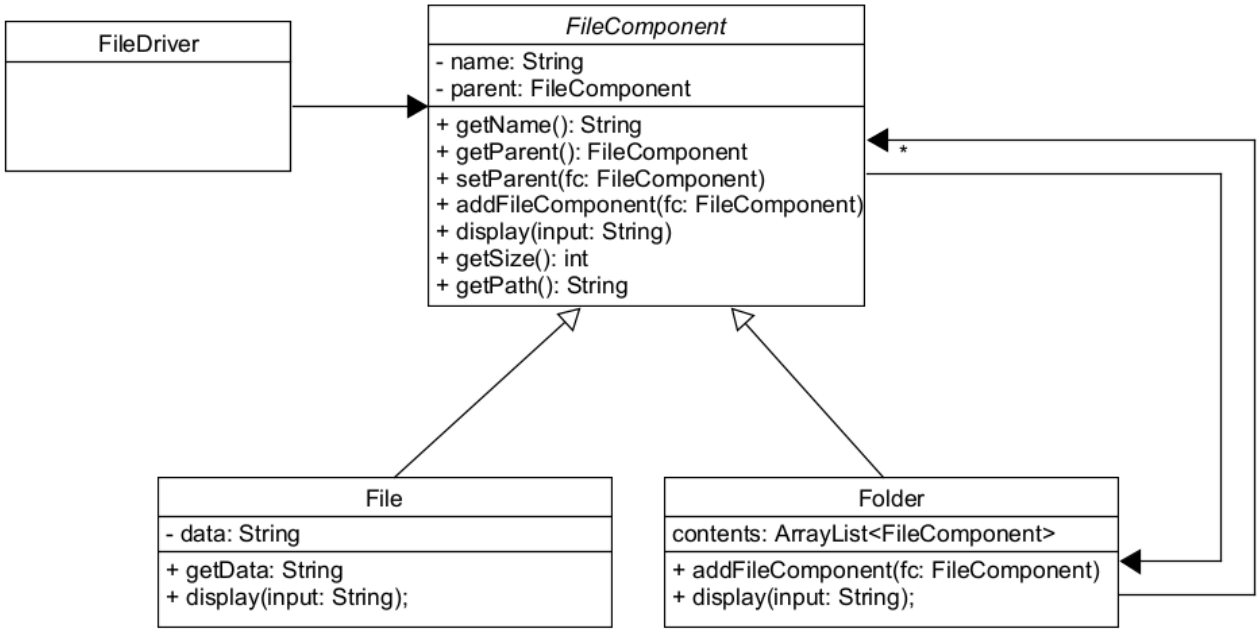
Use the `assignment-4-<your_github_username>` repository on GitHub as the remote repository for Assignment 4.

Refer to the Workflow tutorial section of the Git and GitHub fundamentals document to clone the repo to your local.

PART A

In your Assignment 4 repository is a simulation of a file system that implements the **Composite** pattern. Run the FileDriver to see how the sample data is organized.

Class diagram:



1. Create an object diagram representing objects at the end of execution of main() (FileDriver.java)
2. Complete the implementation of:
   - `getSize()`
     - returns the size of the file
     - returns the total size of all contents of the folder
   - `getPath()` which should work the same for both File and Folder
     e.g.
     `Courses/ACS-3916 HCI/Project/ProjectFinal/Presentation.txt`

PART B

New features! You will now add the ability to back up any item in the file system and search for a keyword. Use the **Visitor** pattern to implement the backup and search functionalities.

- The backup simulation will display the files as they are being backed up i.e. for the Project folder:

```
Backing up folder Project contents:
Backing up file Proposal.txt...
Backing up file MilestoneI.txt...
Backing up file MilestoneII.txt...
Backing up folder ProjectFinal contents:
Backing up file FinalReport.txt...
Backing up file Presentation.txt…
```

- The search simulation will display the path of the files where the keyword is found in the data or filename
- E.g.:
  Search term: *assignment*

```
Courses/ACS-3913 Software Des & Arch/Assignment1.txt
Courses/ACS-3913 Software Des & Arch/Assignment2.txt
Courses/ACS-3913 Software Des & Arch/Assignment3.txt
Courses/ACS-3913 Software Des & Arch/Assignment4.txt
Courses/ACS-3913 Software Des & Arch/Assignment5.txt
Courses/ACS-3916 HCI/Assignment1.txt
Courses/ACS-3916 HCI/Assignment2.txt
```

  Search term: *prototype*

```
Courses/ACS-3916 HCI/Project/MilestoneII.txt
Courses/ACS-3916 HCI/Project/ProjectFinal/FinalReport.txt
```

1. Produce the following design diagrams:
    a) The new class diagram that includes the Visitor pattern for backup and search functions described above
        - Include all changes that were made to the class diagram on page 1
    b) The sequence diagram for performing a search: include the instantiation of SearchVisitor

2. Implement your design.

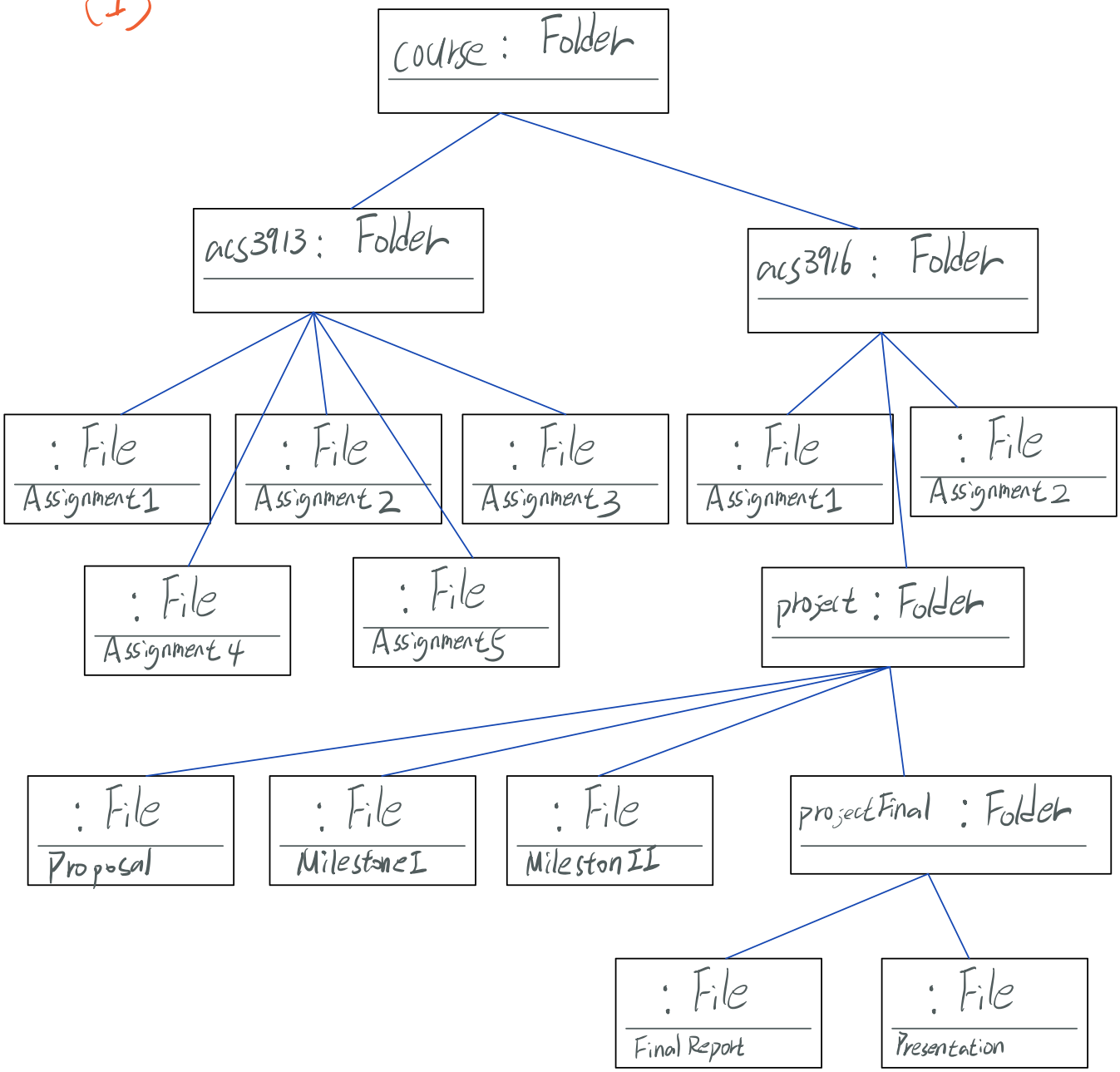
    Your code should work with the test driver provided in your repo.

Submit all your files via GitHub Classroom. Parts A and B will be in the same folder.

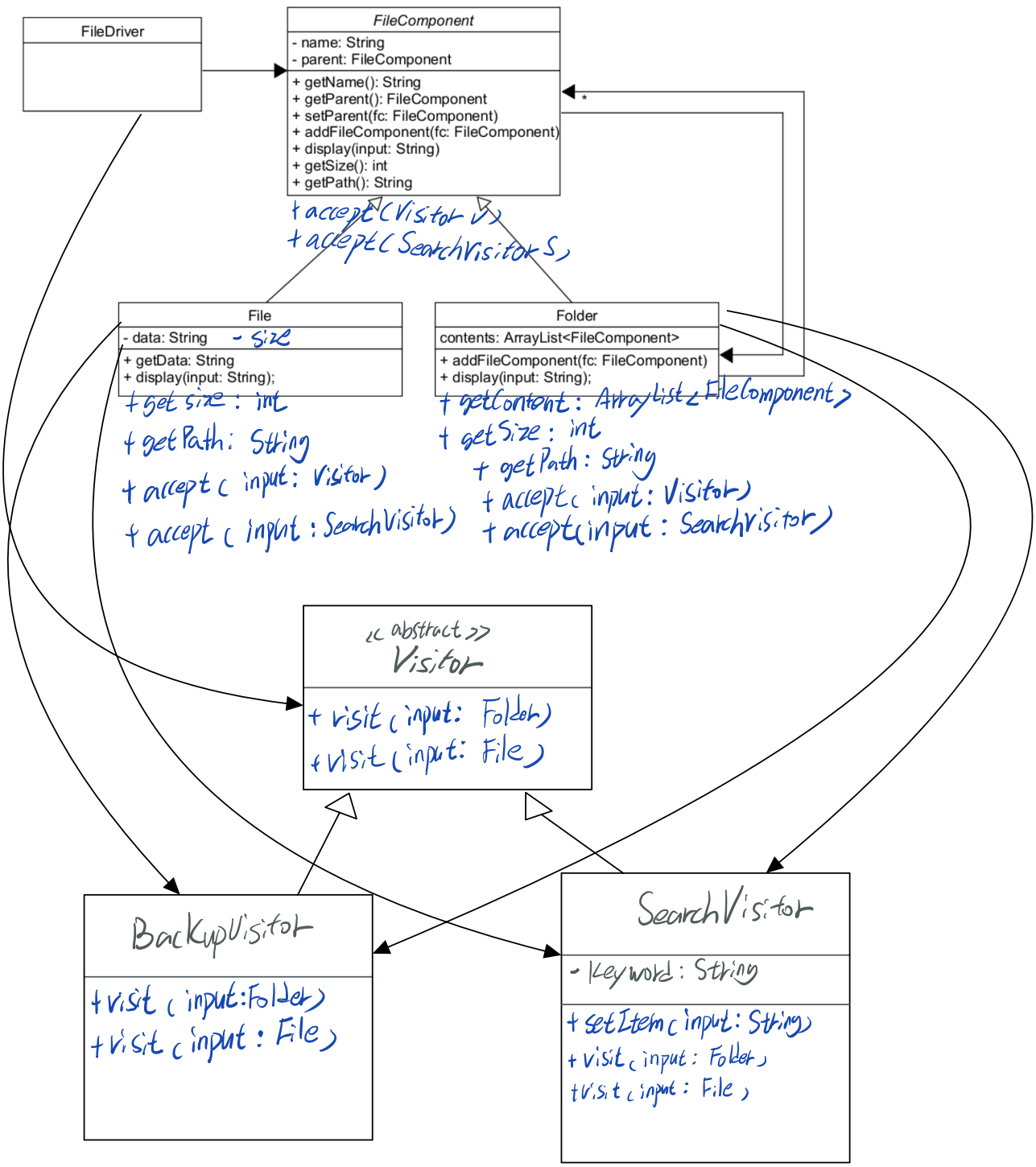
Note that all diagrams must be submitted as PDFs.

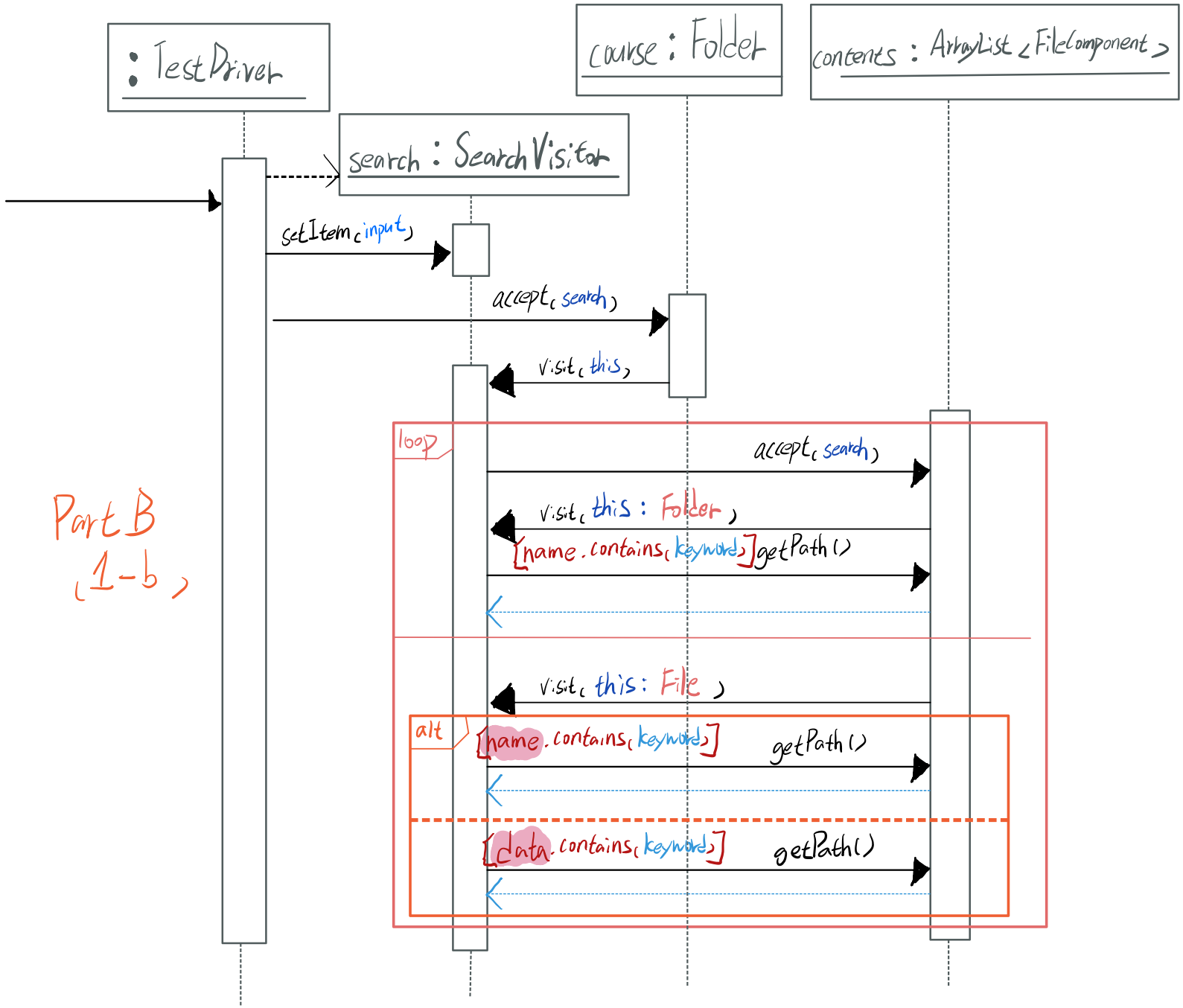
Part A

(1)


course : Folder
acs3913 : Folder
acs3916 : Folder
: File
Assignment1
: File
Assignment2
: File
Assignment3
: File
Assignment1
: File
Assignment2
: File
Assignment4
: File
Assignment5
project : Folder
: File
Proposal
: File
MilestoneI
: File
MilestonII
projectFinal : Folder
: File
Final Report
: File
Presentation

Part B (1-a)


FileDriver
FileComponent
- name: String
- parent: FileComponent
+ getName(): String
+ getParent(): FileComponent
+ setParent(fc: FileComponent)
+ addFileComponent(fc: FileComponent)
+ display(input: String)
+ getSize(): int
+ getPath(): String
+ accept(Visitor v)
+ accept(SearchVisitor s)
File
- data: String
- size
+ getData: String
+ display(input: String);
+ getSize : int
+ getPath : String
+ accept (input : Visitor)
+ accept (input : SearchVisitor)
Folder
contents: ArrayList<FileComponent>
+ addFileComponent(fc: FileComponent)
+ display(input: String);
+ getContent : ArrayList<FileComponent>
+ getSize : int
+ getPath : String
+ accept(input : Visitor)
+ accept(input : SearchVisitor)
<< abstract >>
Visitor
+ visit (input: Folder)
+ visit (input: File)
BackupVisitor
+ visit (input:Folder)
+ visit (input : File)
SearchVisitor
- keyword : String
+ setItem (input : String)
+ visit (input : Folder)
+ visit (input : File)


: TestDriver
course : Folder
contents : ArrayList<FileComponent>
search : SearchVisitor
setItem(input)
accept(search)
visit(this)
loop
accept(search)
visit(this : Folder)
[name.contains(keyword)]getPath()
visit(this : File)
alt
[name.contains(keyword)]
getPath()
[data.contains(keyword)]
getPath()

Part B
(1-b)