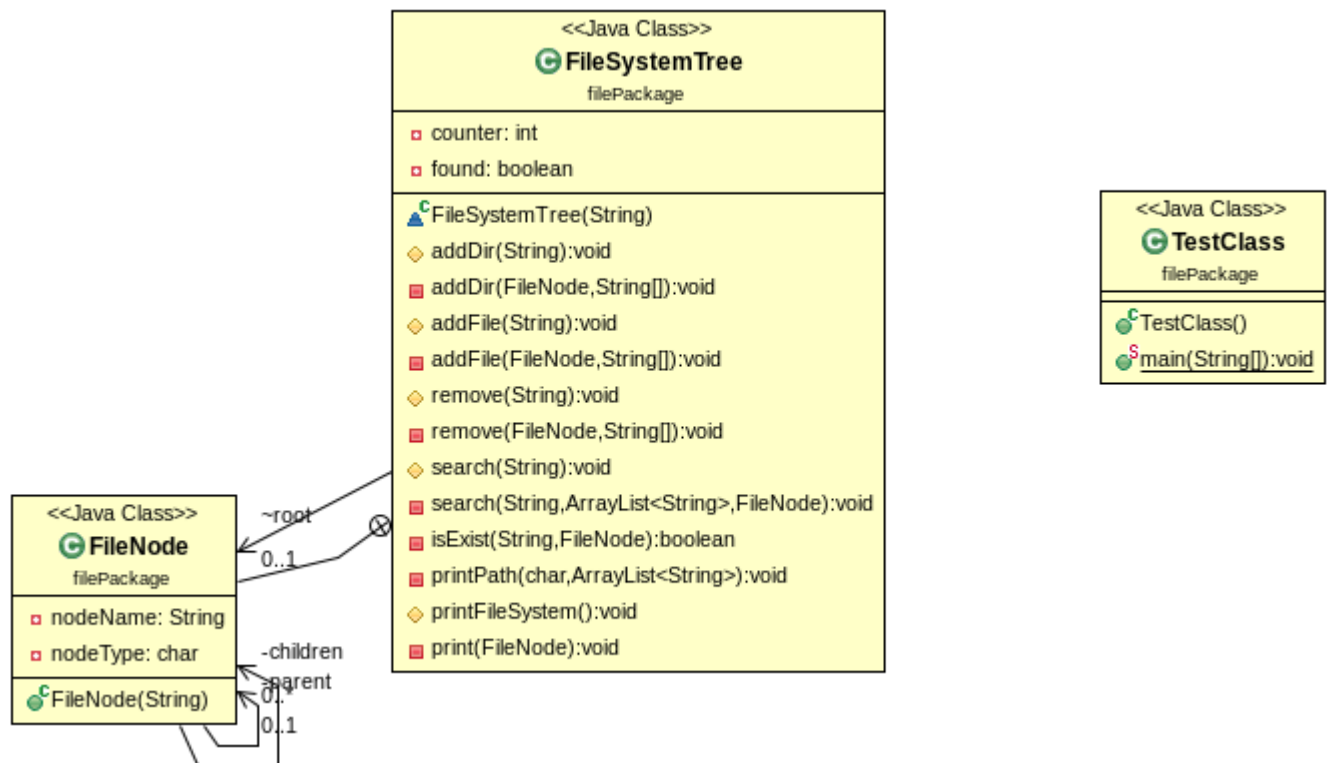


DATA STRUCTURE AND  
ALGORITHMS  
HW05

MOHAMMAD ASHRAF YAWAR  
161044123

# PART01

## CLASS DIAGRAM:



## PROBLEM SOLUTION APPROACH:

- we are asked to implement a general tree structure in java:

I have a main class called `FileSystemTree` and this class is entire tree actually each file or directory in filesystem tree is represented as `FileNode` class which is a nested class of `FileSystemTree` class with private signature.

I represented the relationships between a parent node (which in this case is directory type input) will have several children which simply represented as `ArrayList` data structure.

Each node in arraylist data strudture indicates a child node of a particular parent node where the child nodes can eighther be a file or directory.a FileNode class represents a file or a directory accordigly and each node or directoty keeps it's related sub directories as a list and the name of the node and type of it (if file then nodetype = 'f' if directory then nodetype='d').

I used recursion in each function :

```
>> addDir()
>>addFile()
>>remove()
>>search()
>>some private function
```

all the FileSystemTree class methos are protected so that can only be access through same packages(excluding some private function which area only used inside the FileSystemTree class).

I used ArrayList data structure to maintain easiness an accessibility.

## TEST CASES:

Test Case ID	Test Scenarios	Test Data	Expected Results	Actual Result	Pass/Fail
T01	Add directory	addDir(root/ first_directory)	true	true	Pass
T02	Add file	addFile(root/ first_directory/ new_file.txt)	true	true	Pass
T03	Adding directory into a txt file	Add File(root/first_directory/new_file.txt/ dir)	false	false	Pass
T04	Printing general tree	printFileSystem()	Should print general tree	Prints as expected	Pass
T05	Searching for a key String in generla tree	search("txt")	Should list all the nodes which contains txt string in it	Lists as expected	Pass
T06	Searching for a key String in generla tree	search("---")	Should print blank	Prints empty list	Pass

T07	Removing from tree	remove("root/ first_directory")	It should ask for user if to delete the first_directory or not because first_directory has some sub directories in it	As expected	Pass
T08	Removing from tree	remove("root/ second_directory/ new_directory/ new_file.doc")	Is should delete new_file.doc from general tree.	As expected	Pass

## COMMAND LINE SCREEN SHOTS:

```

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
part01 > src > filePackage > TestClass
Run: TestClass x
CANNOT ADD TO A FILE MAKE SURE YOU SELECT A DIRECTORY TO ADD THE FILE !!!
Printing Tree...
root
--first_directory
--new_file.txt
--second_directory
--new_directory
--new_file.doc
--3th_directory
--file1.txt
--file2.txt

Testing search() ...
file - /root/first_directory/new_file.txt
file - /root/3th_directory/file1.txt
file - /root/3th_directory/file2.txt
Testing remove() ...
this directory has :

first_directory
--new_file.txt

do you still want to delete this directory ? (1) for YES or (0) for NO :0
You selected NO ...
file Removed Successfully !!!
Printing Tree...
root
--first_directory
--new_file.txt
--second_directory
--new_directory
--3th_directory
--file1.txt
--file2.txt

Process finished with exit code 0

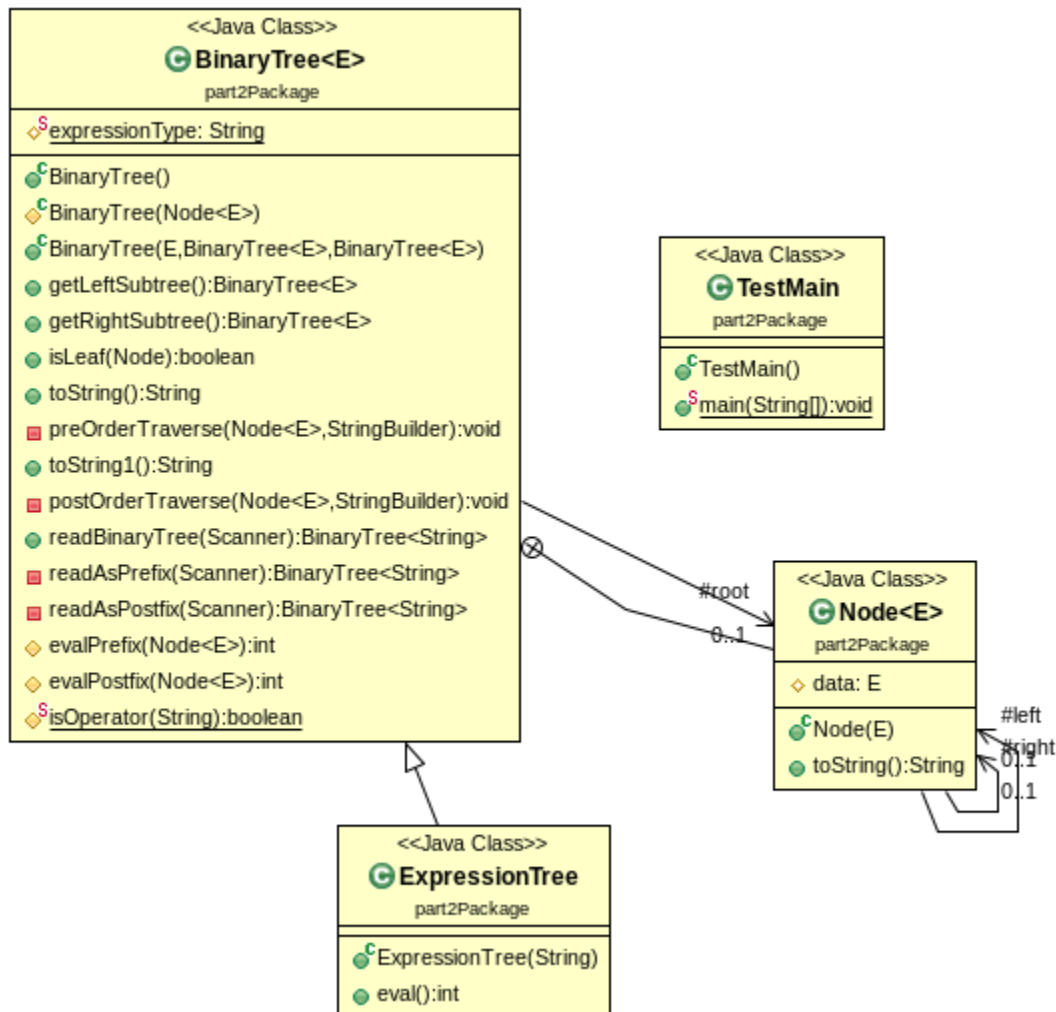
```

```
FileSystemTree [-/Desktop/6.SEMESTER/(1)Data Structures (Bil 222)/HOMEWORKS/HW05/part01] - .../src/filePackage/TestClass.java
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
part01 > src > filePackage > TestClass >
Run: TestClass x
...
10 /usr/lib/jvm/java-11.0-openjdk-amd64/bin/java -javaagent:/opt/idea-IU-193.6911.18/lib/idea_rt.jar=36083:/opt/idea-IU-193.6911.18/bin -Dfile.encoding=UTF-8 -classpath "/home/ashraf/Desktop/6.SEMESTER/(1)Data Structures (Bil 222)
11 /HOMEWORKS/HW05/part01/out/production/part01:/home/ashraf/.m2/repository/org/jetbrains/annotations/18.0.0/annotations-18.0.0.jar" filePackage.TestClass
12 ***** FILE SYSTEM TREE PROJECT *****
13 creating root directory named 'root' ...
14
15 DIRECTORY WITH THE SAME NAME IS ALREADY EXIST !!!
16 CANNOT ADD TO A FILE NAME SURE YOU SELECT A DIRECTORY TO ADD THE FILE !!!
17 Printing Tree...
18 root
19 --first_directory
20 --new_file.txt
21 --second_directory
22 --new_directory
23 --new_file.doc
24 --3th_directory
25 --file1.txt
26 --file2.txt
27
28 Testing search() ...
29 file - /root/first_directory/new_file.txt
30 file - /root/3th_directory/file1.txt
31 file - /root/3th_directory/file2.txt
32 Testing remove() ...
33 this directory has :
34
35 first_directory
36 --new_file.txt
37
38 do you still want to delete this directory ? (1) for YES or (0) for NO :0
39 You selected NO ...
40 file Removed Successfully !!!
41 Printing Tree...
42 root
43 --first_directory
44 --new_file.txt
45 --second_directory
46 --new_directory
47 --3th_directory
48
49 Externally added files can be added to Git
50 View Files Always Add Don't Ask Again
```

```
FileSystemTree [-/Desktop/6.SEMESTER/(1)Data Structures (Bil 222)/HOMEWORKS/HW05/part01] - .../src/filePackage/TestClass.java
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
part01 > src > filePackage > TestClass >
Run: TestClass x
...
10 /HOMEWORKS/HW05/part01/out/production/part01:/home/ashraf/.m2/repository/org/jetbrains/annotations/18.0.0/annotations-18.0.0.jar" filePackage.TestClass
11 ***** FILE SYSTEM TREE PROJECT *****
12 creating root directory named 'root' ...
13
14 Printing Tree...
15 root
16 --first_directory
17 --new_file.txt
18 --second_directory
19 --new_directory
20 --new_file.doc
21 --3th_directory
22 --file1.txt
23 --file2.txt
24
25 Testing search() ...
26 file - /root/first_directory/new_file.txt
27 file - /root/3th_directory/file1.txt
28 file - /root/3th_directory/file2.txt
29 Testing remove() ...
30 this directory has :
31
32 first_directory
33 --new_file.txt
34
35 do you still want to delete this directory ? (1) for YES or (0) for NO :1
36 file Removed Successfully !!!
37 Printing Tree...
38 root
39 --second_directory
40 --new_directory
41 --3th_directory
42 --file1.txt
43 --file2.txt
44
45 Key Promoter X
46 Command Run pressed 925 time(s)
47 'Shift+F10' (Don't show again)
48
49 Externally added files can be added to Git
50 View Files Always Add Don't Ask Again
51 Process finished with exit code 0
```

# PART02

## CLASS DIAGRAM:



## PROBLEM SOLUTION APPROACH:

in this part of project I have used some wrapper functions in order to distinguish between the prefix expression and its evaluations as well as postfix expression and its evaluation.

Since we extended BinaryTree class in ExpressionTree class we can use BinaryTree class and overrides its methods as needed.

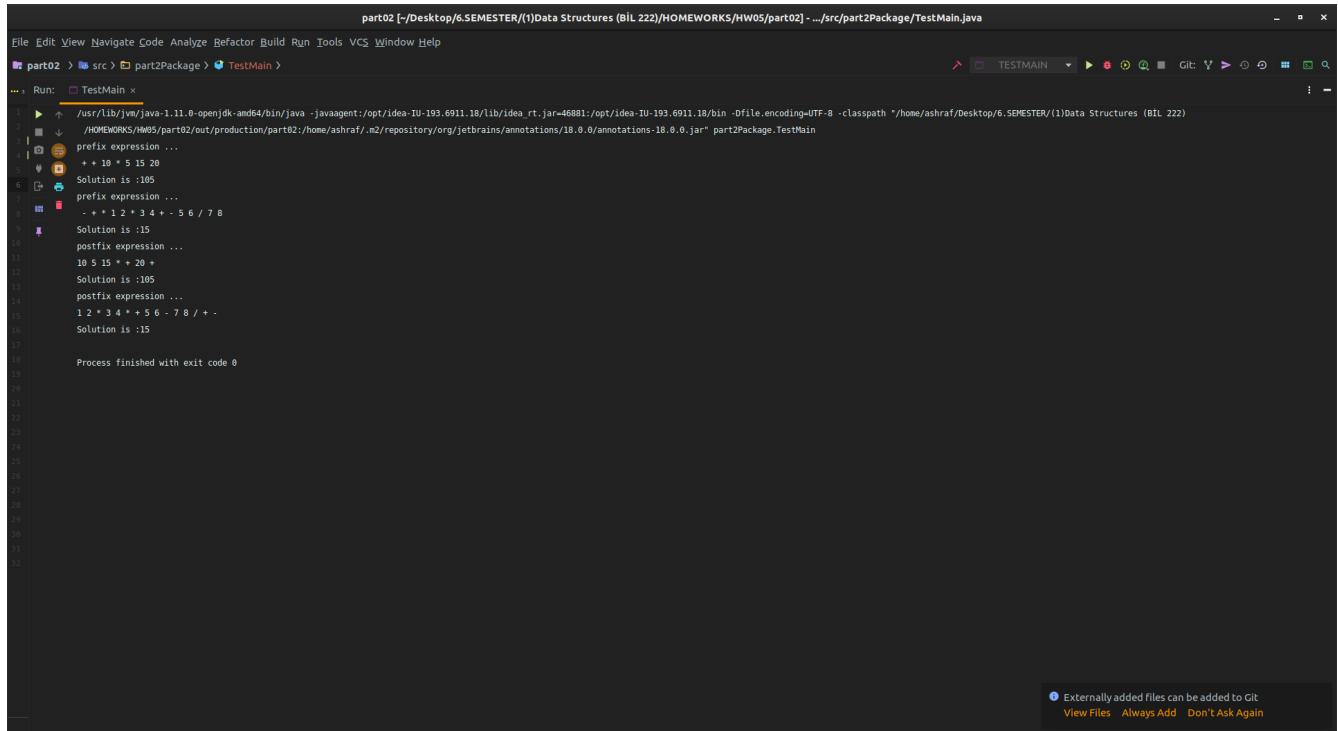
I overrided readBinaryTree class by making it work for both prefix and postfix expressions and construct the certain expression tree according to expression type.

Every necessary methods in each classes has it's own related exception handling part as needed.

I kept root node in BinaryTree class and access it as needed from ExpressionTree class since ExpressionTree class extends BinaryTree class.

Each part of this project has javaDoc related folders.

# COMMAND LINE SCREEN SHOTS:



```
part02 [-/Desktop/6.SEMESTER/(1)Data Structures (BIL 222)/HOMEWORKS/HW05/part02] - .../src/part2Package/TestMain.java
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
part02 > src > part2Package > TestMain >
Run: TestMain x
  /usr/lib/jvm/java-1.11.0-openjdk-amd64/bin/java -javaagent:/opt/idea-IU-193.6911.18/lib/idea_rt.jar-46881:/opt/idea-IU-193.6911.18/bin -Dfile.encoding=UTF-8 -classpath "/home/ashraf/Desktop/6.SEMESTER/(1)Data Structures (BIL 222)/HOMEWORKS/HW05/part02/out/production/part02:/home/ashraf/.m2/repository/org/jetbrains/annotations/18.0.0/annotations-18.0.0.jar" part2Package.TestMain
  prefix expression ...
  ++ 10 * 5 15 20
  Solution is :105
  prefix expression ...
  - + * 1 2 * 3 4 + - 5 6 / 7 8
  Solution is :15
  postfix expression ...
  10 5 15 * + 20 +
  Solution is :105
  postfix expression ...
  1 2 * 3 4 * + 5 6 - 7 8 / + -
  Solution is :15
  Process finished with exit code 0
  Externally added files can be added to Git.
  View Files Always Add Don't Ask Again
```

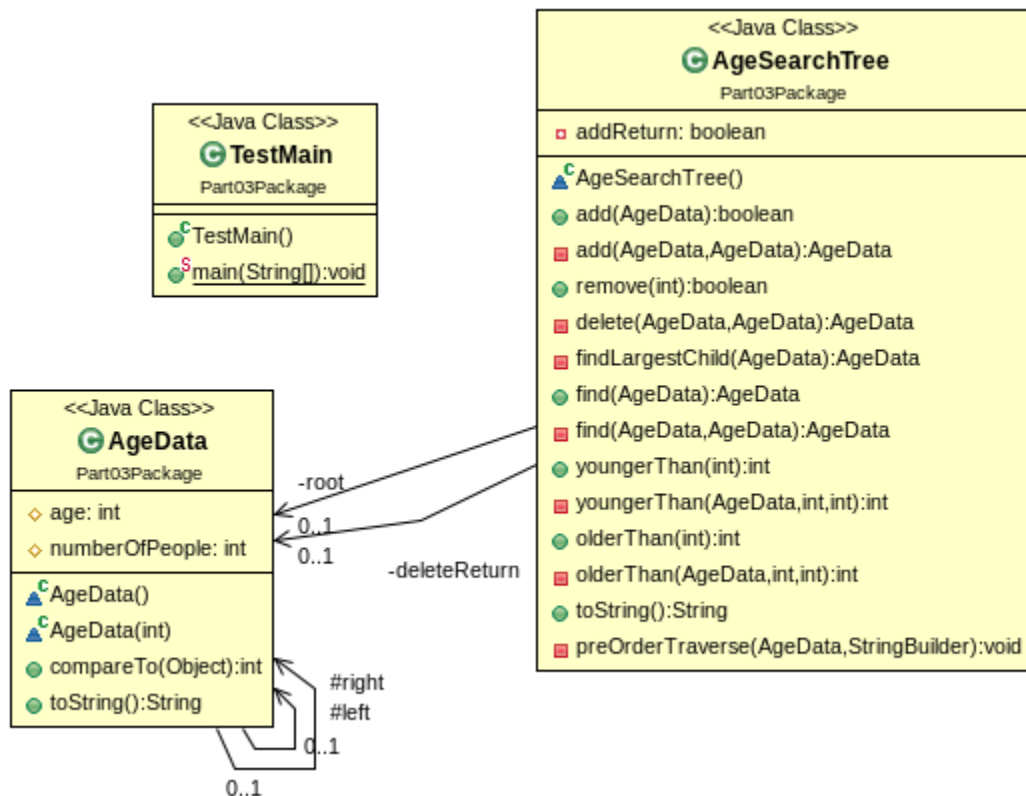


## TEST CASES:

Test Case ID	Test Scenarios	Test Data	Expected Results	Actual Result	Pass/Fail
T01	Construct Expression tree	<code>expTree = new ExpressionTree("+ 10 * 5 15 20");</code>	It should construct a binary expression tree in prefix manner	As expected	Pass
T02	Construct Expression tree	<code>expTree1 = new ExpressionTree("10 5 15 * + 20 +")</code>	It should construct a binary expression tree in postfix manner	As expected	Pass
T03	Print tree using toString()	<code>expTree.toString()</code>	It should print expression tree in preorder manner	As expected	Pass
T04	Print tree using toString1()	<code>expTree1.toString1()</code>	It should print expression tree in postorder manner	As expected	Pass
T05	Evaluate preorder expression tree	<code>ExpTree.eval()</code>	75	75	Pass
T06	Evaluate postorder expression tree	<code>ExpTree1.eval()</code>	75		Pass

# PART03

## CLASS DIAGRAM:



# PROBLEM SOLUTION APPROACH:

– I have used and override methods from BinarySearchTree class which are:

```
add()
delete()
toString()
find()
findLargestChild()
```

– and more additional functions :

```
youngerThan()
olderThan()
preOrderTraverse()
```

- each function has it's own wrapper function to traverse AgeSearchTree recursively .
- AgeSearchTree class uses AgeData class as elements in a BinarySearchTree.
- function who are used internally has private access modifiers .

# COMMAND LINE SCREEN SHOTS:

The image displays two screenshots of an IDE window titled "part03 [-/Desktop/6.SEMESTER/(1)Data Structures (BIL 222)/HOMEWORKS/HW05/part03] - .../src/Part03Package/AgeSearchTree.java". The IDE interface includes a menu bar (File, Edit, View, Navigate, Code, Analyze, Refactor, Build, Run, Tools, VCS, Window, Help), a toolbar with icons for running tests, and a "Run" configuration dropdown set to "TESTMAIN".

The first screenshot shows the command line output for the "Testing add() ..." test. The output displays the execution of the Java program, including the classpath and the main method call.

The second screenshot shows the command line output for the "Testing olderThan() for value 15 ..." test. The output displays the execution of the Java program, including the classpath and the main method call. The output also shows the results of the "Testing delete() first Attempt ..." and "Testing delete() second Attempt ..." tests, indicating that the delete operation was successful.

```

part03 [-/Desktop/6.SEMESTER/(1)Data Structures (BIL 222)/HOMEWORKS/HW05/part03] - .../src/Part03Package/AgeSearchTree.java
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
part03 > src > Part03Package > AgeSearchTree >
Run: TestMain x
Testing...
10 - 1
5 - 1
1 - 1
0 - 1
null
null
null
null
20 - 1
15 - 1
null
null
null
Testing delete() second Attempt ...
Testing ...
5 - 1
1 - 1
0 - 1
null
null
null
20 - 1
15 - 1
null
null
null
*****
Testing find() ...
Expressed output 5 - 1
5 - 1
*****
Process finished with exit code 0

```

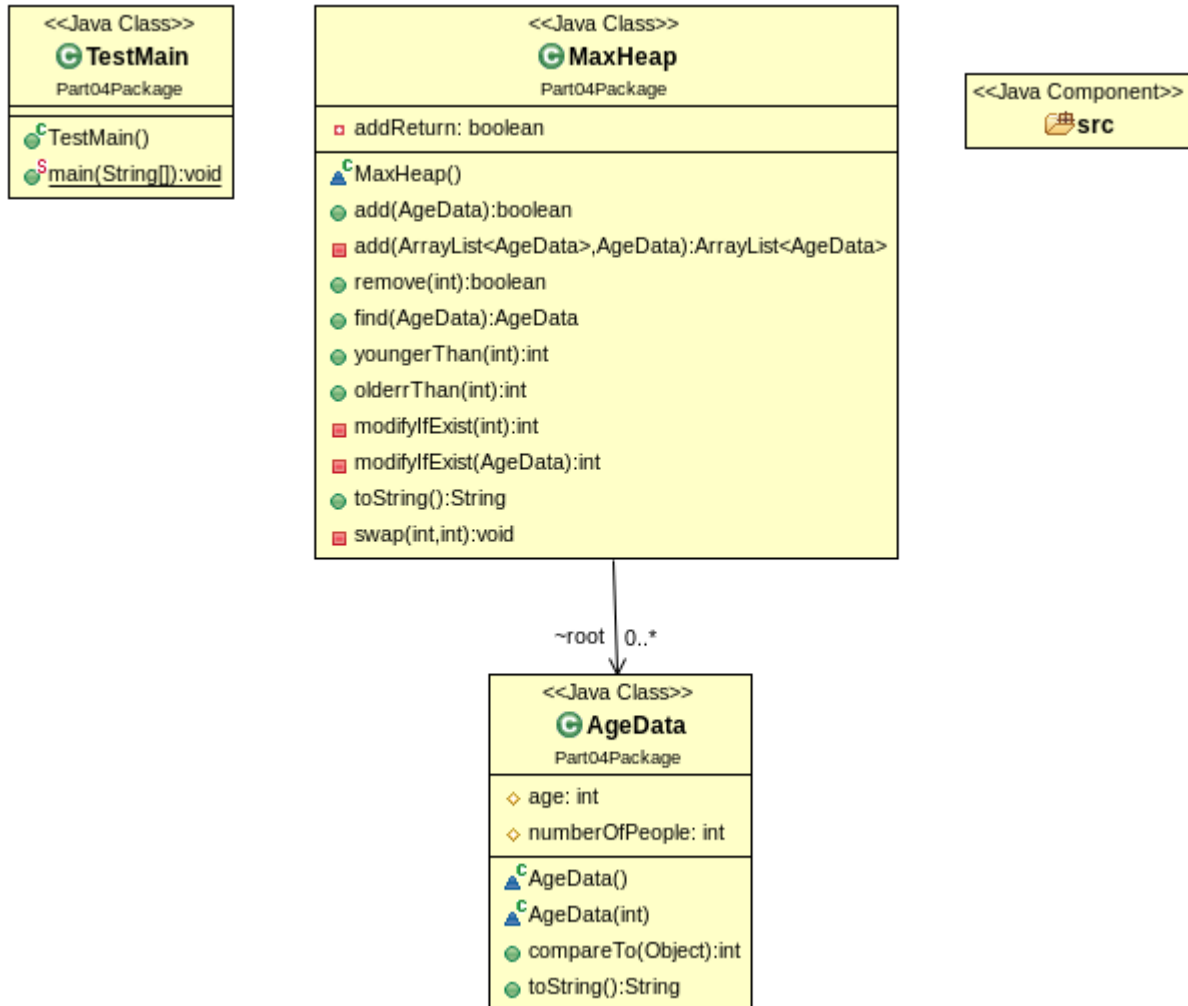
# TEST CASES:

Test Case ID	Test Scenarios	Test Data	Expected Results	Actual Result	Pass/Fail
T01	Add an age to BinarySearchTree	ageTree.add(new AgeData(10))	It should as 10 to tree	As expected	Pass
T02	Add an age to BinarySearchTree	ageTree.add(new AgeData(20))	It should as 20 to tree	As expected	Pass
T03	Add an age to BinarySearchTree	ageTree.add(new AgeData(20))	It should as 20 to tree	As expected	Pass
T04	Add an age to BinarySearchTree	ageTree.add(new AgeData(20))	It should as 20 to tree	As expected	Pass
T05	Add an age to BinarySearchTree	ageTree.add(new AgeData(10))	It should as 10 to tree	As expected	Pass
T06	Add an age to BinarySearchTree	ageTree.add(new AgeData(1))	It should as 1 to tree	As expected	Pass
T07	Print tree using toString	System.out.println(ageTree.toString())	It should print tree	As expected	Pass

T08	Find counts of the age in tree which are smaller than the given input age(youngerThan)	ageTree.youngerThan(5)	1	1	Pass
T09	Find counts of the age in tree which are smaller than the given input age(youngerThan)	ageTree.youngerThan(10)	2	2	Pass
T10	Find counts of the age in tree which are smaller than the given input age(youngerThan)	ageTree.youngerThan(15)	3	3	Pass
T11	Find counts of the age in tree which are bigger than the given input age(olderThan)	ageTree.olderThan(5))	2	2	Pass
T12	Find counts of the age in tree which are bigger than the given input age(olderThan)	ageTree.olderThan(10))	1	1	Pass
T13	Find counts of the age in tree which are bigger than the given input age(olderThan)	ageTree.olderThan(15))	1	1	Pass
T14	Removing an element which is in the tree	remove(10)	Should returns the object if removed	As expected	Pass
T15	Removing an element which is NOT in the tree	remove(100)	Should return null	As expected	Pass
T16	Finding an existing element in the tree	find(new AgeData(20))	It should return corresponding age and number of people node	As expected	Pass
T17	Finding NON existing element in the tree	find(new AgeData(100))	It should return false	As expected	Pass

# PART04

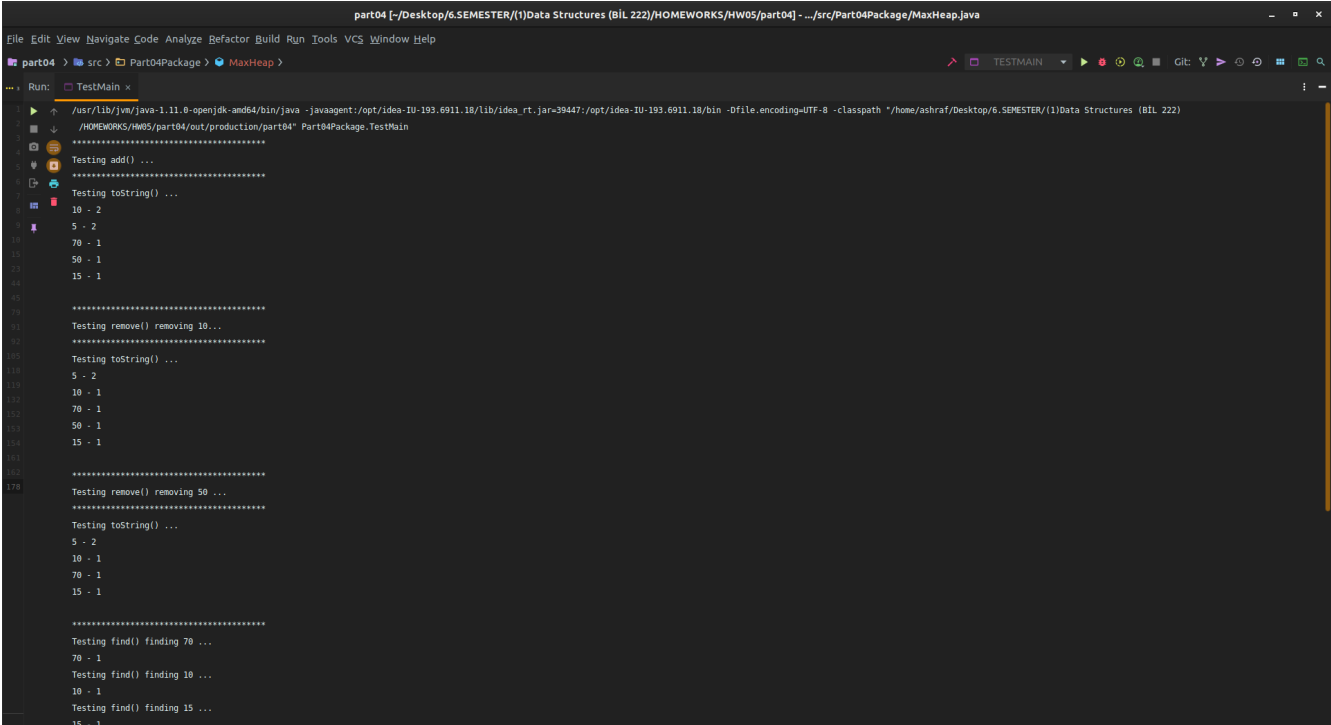
## CLASS DIAGRAM:



# PROBLEM SOLUTION APPROACH:

- I have used compareTo methods to compare the numberOfPeople in any particular node of heap tree.
- heap class which holds an ArrayList to represent heap tree (max\_heap tree) and comparison is done considering counts of people in a particular node of heap elements.
- AgeData class is used to hold the information about one age and its count

# COMMAND LINE SCREEN SHOTS:



```
part04 [-/Desktop/6.SEMESTER(1)Data Structures (BIL 222)/HOMEWORKS/HW05/part04] - .../src/Part04Package/MaxHeap.java
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
part04 > src > Part04Package > MaxHeap >
Run: TestMain x
  /usr/lib/jvm/java-11.0-openjdk-amd64/bin/java -javaagent:/opt/idea-IU-193.6911.18/lib/idea_rt.jar-39447:/opt/idea-IU-193.6911.18/bin -Dfile.encoding=UTF-8 -classpath "/home/ashraf/Desktop/6.SEMESTER(1)Data Structures (BIL 222)
  /HOMEWORKS/HW05/part04/out/production/part04" Part04Package.TestMain
  *****
  Testing add() ...
  *****
  Testing toString() ...
  10 - 2
  5 - 2
  70 - 1
  50 - 1
  15 - 1
  *****
  Testing remove() removing 10...
  *****
  Testing toString() ...
  5 - 2
  10 - 1
  70 - 1
  50 - 1
  15 - 1
  *****
  Testing remove() removing 50 ...
  *****
  Testing toString() ...
  5 - 2
  10 - 1
  70 - 1
  15 - 1
  *****
  Testing find() finding 70 ...
  70 - 1
  Testing find() finding 10 ...
  10 - 1
  Testing find() finding 15 ...
  15 - 1
```

```
part04 [-/Desktop/6.SEMESTER/1)Data Structures (BIL 222)/HOMEWORKS/HW05/part04] - .../src/Part04Package/MaxHeap.java
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
part04 > src > Part04Package > MaxHeap >
Run: TestMain x
1 15 - 1
2 *****
3 Testing remove() removing 50 ...
4 *****
5 Testing toString() ...
6 5 - 2
7 10 - 1
8 70 - 1
9 15 - 1
10
11 *****
12 Testing find() finding 70 ...
13 70 - 1
14 Testing find() finding 10 ...
15 10 - 1
16 Testing find() finding 15 ...
17 15 - 1
18 *****
19 Testing youngerThan()...
20 Testing with value 5 ...
21 0
22 Testing with value 50 ...
23 3
24 Testing with value 15 ...
25 2
26 *****
27 Testing olderThan()...
28 Testing with value 5 ...
29 3
30 Testing with value 50 ...
31 1
32 Testing with value 15 ...
33 1
34 *****
35
36 Process finished with exit code 0
37
38 Externally added files can be added to Git.
39 View Files Always Add Don't Ask Again
```

## TEST CASES:

Test Case ID	Test Scenarios	Test Data	Expected Results	Actual Result	Pass/Fail
T01	Adding element into heap tree	add(new AgeData(10))	It should add 10 to heap tree	As expected	Pass
T02	Adding element into heap tree	add(new AgeData(5))	It should add 5 to heap tree	As expected	Pass
T03	Adding element into heap tree	add(new AgeData(70))	It should add 70 to heap tree	As expected	Pass
T04	Adding element into heap tree	add(new AgeData(10))	It should increment number of people of node which holds age 10 and then update the heap tree	As expected	Pass
T05	Adding element into heap tree	add(new AgeData(50))	It should add 50 to heap tree	As expected	Pass



T06	Adding element into heap tree	add(new AgeData(5))	It should increment number of people of node which holds age 5 and then update the heap tree	As expected	Pass
T07	Adding element into heap tree	add(new AgeData(15))	It should add 15 to heap tree	As expected	Pass
T08	Displaying heap tree using toString method	heap.toString()	It should retruns string holding heap tree to be displayed using print methods of java	As expected	Pass
T09	Removing an element existing in heap tree	heap.remove(10);	It should decrement the number of people of node which holds 10 since our 10 counts were 2 and then update the heap tree	As expected	Pass
T10	Removing an element existing in heap tree	heap.remove(50);	It should delete node which holds 50 in it since we have only one 50 so it will delete the corresponding node	As expected	Pass
T11	Finding an existing element in the heap tree	find(new AgeData(5))	It should return corresponding age and number of people node	As expected	Pass
T12	Finding NON existing element in the heap tree	find(new AgeData(50))	It should return false	As expected	Pass
T13	Find counts of the age in tree which are smaller than the given input age(youngerThan)	heap.youngerThan(5)	0	0	Pass
T14	Find counts of the age in tree which are smaller than the given input age(olderThan)	heap.olderThan(10)	2	2	Pass

