GIT Department of Computer Engineering

CSE 222 - Spring 2020 Homework-5 Report

Ali BAHAR

171044066

Problem Solution Approach Q-1

First i created the FileNode class to be able to handle the node of the tree. We can create a general tree structure by using this FileNode class. There is three different fields in every node. One of them keeps the name of the node as String, one of them keeps the children nodes as ArrayList, the last one keeps the type of the node. Node type can be file or directory. We can use this FileNode class to add new child or get some information about the node. Then i created the FileSystemTree that can handle a file system hierarchy in a general tree structure. I have used the FileNode class to handle the nodes of this tree. Then i have implemented the required methods. There is three different add method. One of them does the actual adding by using recursive calls and the other methods call this method by specifying the type of the node. Most of the methods split the given path as string to string array.

Problem Solution Approach Q-2

First I implemeted the BinaryTree as described in the book. The ExpressionTree extends from this BinaryTree class. Given expression string may be prefix or postfix so first ı determine the expression is prefix or postfix. Then if it is prefix ,then it is evaluated from begin to end.if it is postfix ,then it is evaluated from end to begin.

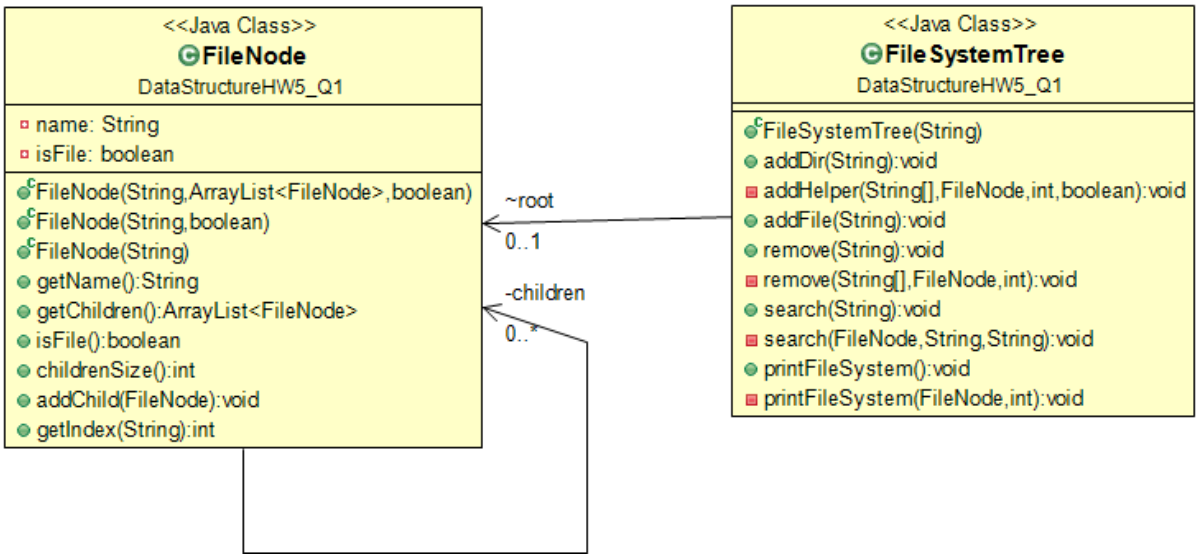
Problem Solution Approach Q-3

First I need a binary search tree class. I have had already a binary tree class from previous part so the binary search tree extends from binary tree class and implements a search tree interface. I have created the binary search tree by getting help from book. My AgeSearchTree class extends from binary search tree class. It is a generic class but it can just keep the data which extends from AgeDataInterface because I need some methods from AgeData class.

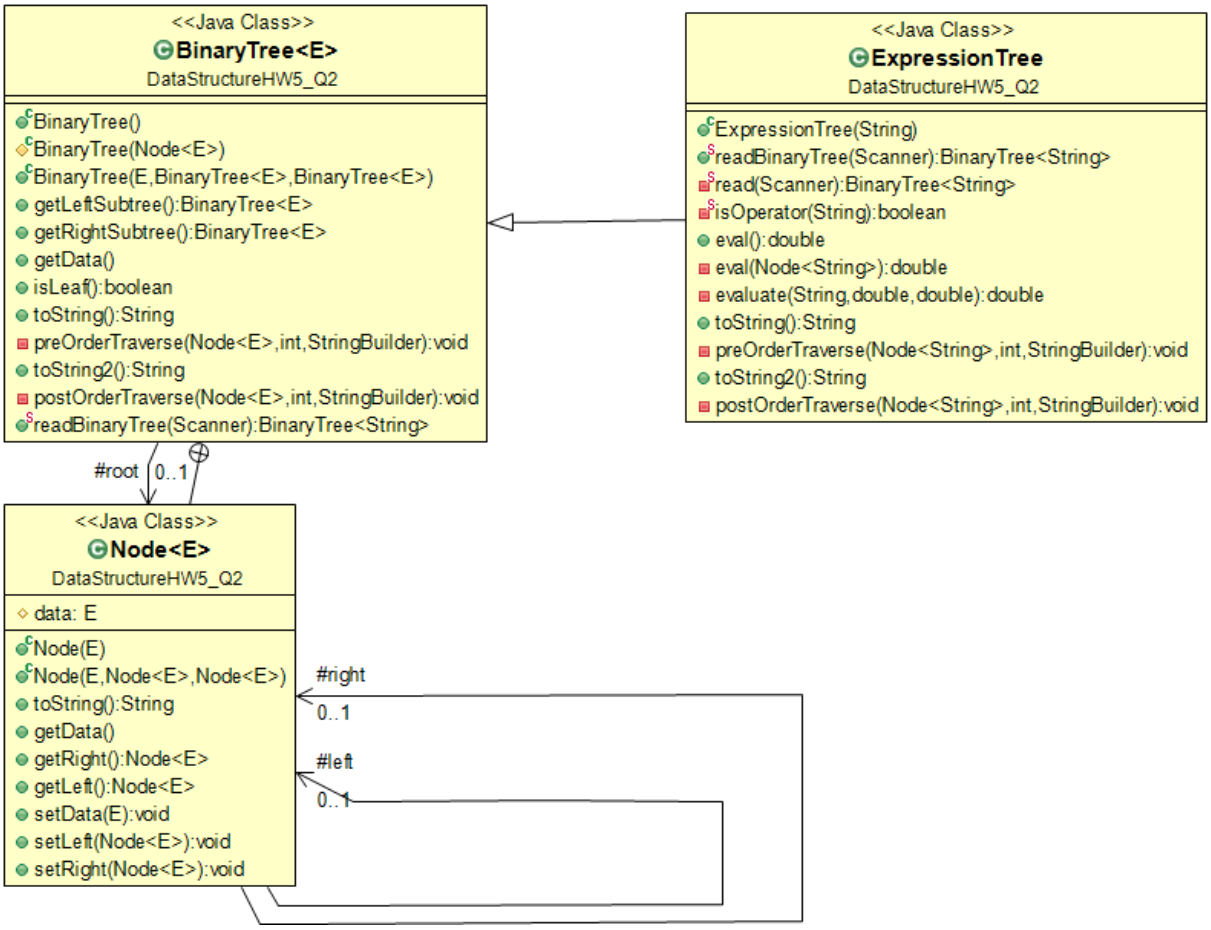
Problem Solution Approach Q-4

The problem was same with question 3 but there was a differences. Question was solve this problem by using max heap. (where the maximum element is in the root node).I have implemented the heap class by using ArrayList as described in homework. This MaxHeap class is a generic class but the data must be extended at least from AgeDataInterface because we need some AgeDataInterface methods. I have used the same AgeData class which comes from question 3. But ı have needed to change the comparison method. Because in this question my heap shoul be constructed by comparising the number of people instead of ages. Therefore I have create a new comparator class and this MaxHeap class have another constructor method that takes a comparator as paramter. It makes more object oriented. Also there is a no-parameter constructor and this creates a default comparator that compares the number of people.

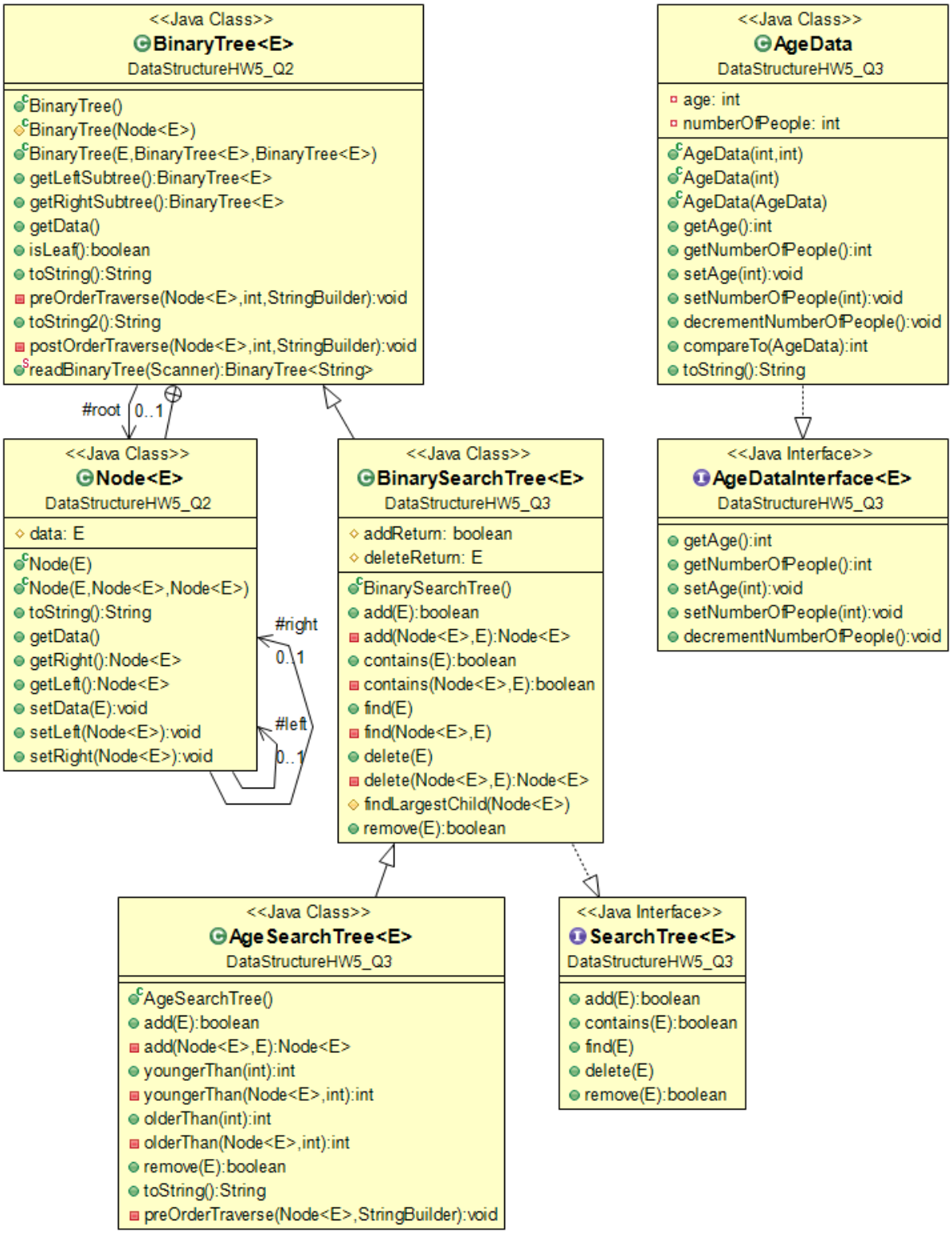
Class Diagram-Q1



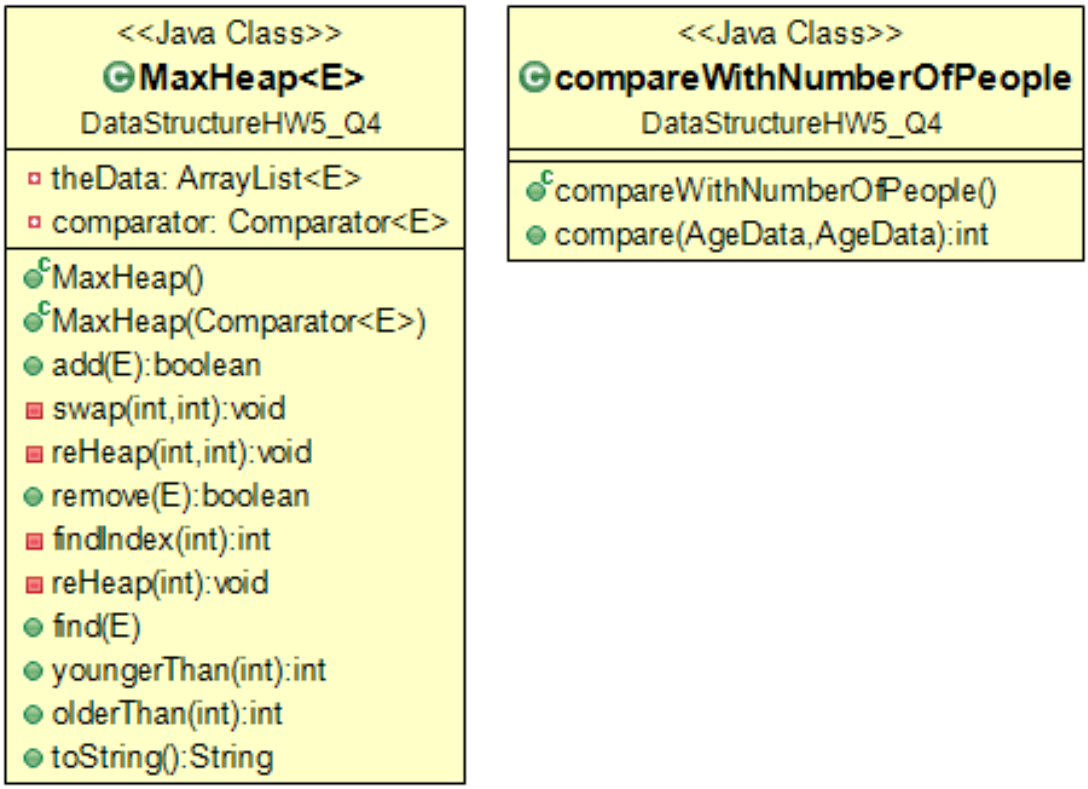
Class Diagram-Q2



Class Diagram-Q3



Class Diagram-Q4



Test Cases-Q1

Test ID	Scenerio	TestData	Expected Result	Actual Result	Pass /Fail
Q1.0	Create a FileSystemTree and fill it by using addDir and addFile methodes.	Valid paths that can be added to FileSystemTree.	No Warning,No Error. Directories and files has been added successfully.	As Expected	Pass
Q1.1	Print the all tree and see that the add methods worked properly.	myFileSystem	All directories and files is printed.	As Expected	Pass
Q1.2	Invalid parameters are passed into add methods.	Some invalid paramters.	There is no new node insde the tree. There is warning because a file node can not have a child node.	As Expected	Pass
Q1.3	Search method tested.	TheWord = “File” TheWord = “A” TheWord = “a” TheWord = “.” (All inputs are valid)	All files and directories which contain this words will be printed with their all path.	As Expected	Pass

Q1.4	Search method tested with invalid parameters.	Invalid parameters.	Nothing will be printed.	As Expected	Pass
Q1.5	Remove some files and directories.and print the all tree.	root/dev/pts/file1.doc root/dev/pts/file2.doc root/dev/pts	Files and directories which specified is removed.	As Expected	Pass
Q1.6	Remove some directories which have other files and directories inside it.	root/home/Ali root/mnt/c	The content of this directories is printed. And asked to user whether to remove or not.	As Expected	Pass
Q1.7	Try to remove a node which doesn't exist.	root/INVALID	Warning.	As Expected	Pass
Q1.7	Try to remove a node which doesn't exist.	Root/home/Ali/ Desktop/file0	Warning.	There is no warning but the file is not removed	Fail
Q1.8	Print the tree	myFileSystem	Printed all tree	As Expected	Pass

Test Cases-Q2

Test ID	Scenario	Steps	Test Data	Expected Results	Actual Result	Pass /Fail
1	Test the constructor	Create four different object.	ExpTree1= "+ + 10 * 5 15 20" ExpTree2= "10 5 15 * + 20 +" ExpTree3 = "* + * 40 3 / 6 42 / + 2 0 –37 67" ExpTree4= "40 3 * 42 6 / + 67 37 – 2 0 + / *"	All objects are created.	As Exprected	Pass
2	Test the toString method.	Print the all objects which created previous test case.	ExpTree1 ExpTree2 ExpTree3 ExpTree4	Tree is printed.	As Exprected	Pass
3	Test the toString2 method	Print the all objects which created previous test case.	ExpTree1 ExpTree2 ExpTree3 ExpTree4	Tree is printed.	As Expected	Pass
4	Test the eval method	Evaluate the expression tree.	ExpTree1 ExpTree2 ExpTree3 ExpTree4	Result is printed	As Expected	Pass

Test Cases-Q3

Test ID	Scenario	Steps	Test Data	Expected Results	Actual Result	Pass /Fail
1	Test the add method of the AgeSearchTree class.	1-add new data 2-add new data 3-add new data 4-add new data 5-add a data which is already exist in this tree. 6-add new data 7-add new data 8-add new data 9-add a data which is already exist in this tree. 10-add a data which is already exist in this tree. 11-add a data which is already exist in this tree. 12-add a data which is already exist in this tree. 13-add a data which is already exist in this tree.	Age = 10 Age = 20 Age = 5 Age = 15 Age = 10 Age = 2 Age = 7 Age = 30 Age = 2 Age = 2 Age = 30 Age = 30 Age = 30	All data has been added to tree.	As Expreced	Pass
2	Test the toString method of the AgeSearchTree class.	1-Create a string representing the tree. 2-print it.	Age tree object which has created previous test.	Prints the all tree.	As Expreced	Pass
3	Test the youngerThan method of the AgeSearchTree class.	1- pass some age to method	anAge = 15 anAge = 30 anAge = 35 anAge = 5 anAge = 1	Prints the number of people younger than given age.	As Expreced	Pass
4	Test the olderThan method of the	1- pass some age to method	anAge = 15 anAge = 29 anAge = 35	Prints the number of people older than given age.	As Expreced	Pass

	AgeSearchTree class.		anAge = 5 anAge = 1			
5	Test the find method	1- pass some age to method 2- call the toString method of returned value.	age = 30 age = 15 age = 0	30 and 15 has been found but 0 couldn't be found so find method returned null. Then NullPointerException has been thrown because we tried to call toString method for null which has returned from find method.	As Expected	Pass
6	Test the remove method of the AgeSearchTree.	1-remove from leaf. 2-remove the node which have one subtree. 3-remove the node which is not in tree. 4-remove a node which have two subtree. 5-remove a node from root(there was two people. after remove there is just one people). 6-remove a node from root. 7-remove a node from leaf(there was more than 1 people.) 8-remove a node from leaf(there was more than 1 people.) 9- remove the node which is not in tree.	Age = 7 Age = 5 Age = 7 Age = 20 Age = 10 Age = 10 Age = 30 Age = 30 Age = 100 Age = 0	If the node which will be removed is in the tree and if number of people is one then this node removed.But the number of people is greater than 1 ,then the number of people is decreased.	As Expected	Pass

		10- remove the node which is not in tree.				

Test Cases-Q4

Test ID	Scenario	Steps	Test Data	Expected Results	Actual Result	Pass /Fail
1	Test the add method of the MaxHeap class.	1-add new data(7 times) 2-add a data which is already exist in this heap.(6 times)	Item =10 Item =20 Item =30 Item =40 Item =50 Item =60 Item =70 Item =10 Item =20 Item =30 Item =40 Item =10 Item = 40	All data has been added to heap.	As Exprected	Pass
2	Test the toString method of MaxHeap.	1 – print the string representation of the heap.	The heap which has been created previous test case.	Prints the all elements of the heap.	As Exprected	Pass
3	Test yougerThan method of the MaxHeap class.	Pass some age to method.	anAge = 15 anAge = 5 anAge = 80 anAge = 40 anAge = 45	Prints the number of people younger than given age.	As Exprected	Pass
4	Test olderThan method of the MaxHeap class.	Pass some age to method.	anAge = 15 anAge = 80 anAge = 5 anAge = 40 anAge = 45	Prints the number of people older than given age.	As Exprected	Pass
5	Test the find method of the MaxHeap class.	1- pass some age to method 2-call the toString method of returned value.	age = 30 age = 70 age = 100	30 and 70 has been found but 100 couldn’t be found so find method returned null. Then NullPointerException has been thrown because we tried to call toString method for null which has returned from find method.	As Exprected	Pass

6	Test the remove method of the MaxHeap class.	1- remove an age. 2-print the heap	Item = 7 Item = 50 Item = 10 Item = 10 Item = 10 Item = 10 Item = 40 Item = 35 Item = 0	If the node which will be removed is in the heap and if number of people is one, then this node removed. But the number of people is greater than 1 ,then the number of people is decreased and reheap.	As Exprecte d	Pass

Running Command and Results-Q1

The tree is created to be tested.

The tree after created :

root

dev

pts

file1.doc

file2.doc

fd

file1.c

file2.c

ttyS1.i

ttyS2.i

ttyS3.i

home

Ali

Desktop

file0.desktop

file1.desktop

file2.desktop

mnt

c

ProgramFiles

programFile1.txt

programFile2.txt

programFile3.txt

programFile4.txt

try to some invalid input for addFile and addDir methods.

file can't have child.

After some invalid inputs :

root

dev

pts

file1.doc

file2.doc

fd

file1.c

file2.c

ttyS1.i

ttyS2.i

ttyS3.i

home

Ali

Desktop

file0.desktop

file1.desktop

file2.desktop

mnt

c

ProgramFiles

programFile1.txt

programFile2.txt

programFile3.txt

programFile4.txt

The search method Tested (theWord = "Pro")

dir - root/mnt/c/ProgramFiles

file - root/mnt/c/ProgramFiles/programFile1.txt

file - root/mnt/c/ProgramFiles/programFile2.txt

file - root/mnt/c/ProgramFiles/programFile3.txt

file - root/mnt/c/ProgramFiles/programFile4.txt

The search method Tested (theWord = "A")

dir - root/home/Ali

The search method Tested (theWord = "a")

dir - root/mnt/c/ProgramFiles

file - root/mnt/c/ProgramFiles/programFile1.txt

file - root/mnt/c/ProgramFiles/programFile2.txt

file - root/mnt/c/ProgramFiles/programFile3.txt

file - root/mnt/c/ProgramFiles/programFile4.txt

The search method Tested (theWord = ".")

file - root/dev/pts/file1.doc

file - root/dev/pts/file2.doc

file - root/dev/fd/file1.c

file - root/dev/fd/file2.c

file - root/dev/ttyS1.i

file - root/dev/ttyS2.i

file - root/dev/ttyS3.i

file - root/home/Ali/Desktop/file0.desktop

file - root/home/Ali/Desktop/file1.desktop

file - root/home/Ali/Desktop/file2.desktop

file - root/mnt/c/ProgramFiles/programFile1.txt

file - root/mnt/c/ProgramFiles/programFile2.txt

file - root/mnt/c/ProgramFiles/programFile3.txt

file - root/mnt/c/ProgramFiles/programFile4.txt

The search method Tested with invalid parameter(theWord = "INVALID")

The search method Tested with invalid parameter(theWord = "ttys4")

Some files and directories has been removed :

try to remove root/dev/pts/file1.doc

file1.doc is removed.

try to remove root/dev/pts/file2.doc

file2.doc is removed.

try to remove root/dev/pts

pts is removed.

After remove operation :

root

dev

fd

file1.c

file2.c

ttyS1.i

ttyS2.i

ttyS3.i

home

Ali

Desktop

file0.desktop

file1.desktop

file2.desktop

mnt

c

ProgramFiles

programFile1.txt

programFile2.txt

programFile3.txt

programFile4.txt

try to remove root/home/Ali

Content of Ali

Desktop

file0.desktop

file1.desktop

file2.desktop

the directory includes some other directories (or files)

Do you want to remove them also.(press y ,if you want to remove)

y

Ali is removed with its content.

try to remove root/mnt/c

Content of c

ProgramFiles

programFile1.txt

programFile2.txt

programFile3.txt

programFile4.txt

the directory includes some other directories (or files)

Do you want to remove them also.(press y ,if you want to remove)

y

c is removed with its content.

After remove operation :

root

dev

fd

file1.c

file2.c

ttyS1.i

ttyS2.i

ttyS3.i

home

mnt

try to remove root/INVALID

WARNING : path cannot be found.

try to remove root/home/Ali/Desktop/file0

The tree after Test:

root

dev

fd

file1.c

file2.c

ttyS1.i

ttyS2.i

ttyS3.i

home

mnt

Running Command and Results-Q2

four different Expression tree object has been created.

expTree1.toString() : + + 10 * 5 15 20

expTree2.toString() : + + 10 * 5 15 20

expTree3.toString() : * + * 40 3 / 42 6 / - 67 37 + 2 0

expTree4.toString() : * + * 40 3 / 42 6 / - 67 37 + 2 0

expTree1.toString2() : 10 5 15 * + 20 +

expTree2.toString2() : 10 5 15 * + 20 +

expTree3.toString2() : 40 3 * 42 6 / + 67 37 - 2 0 + / *

expTree4.toString2() : 40 3 * 42 6 / + 67 37 - 2 0 + / *

Evaluation of expTree1 : 105.0

Evaluation of expTree2 : 105.0

Evaluation of expTree3 : 1905.0

Evaluation of expTree4 : 1905.0

Running Command and Results-Q3

The number of people younger than 35 : 13

The number of people younger than 5 : 3

The number of people younger than 1 : 0

The number of people older than 15 : 5

The number of people older than 30 : 4

The number of people older than 35 : 0

The number of people older than 5 : 9

The number of people older than 1 : 13

find the number of people at 30 : 30 - 4

find the number of people at 15 : 15 - 1

find the number of people at 0 :

NullPointerException has been thrown because find method returned null.

7 is removed.

5 is removed.

7 is not removed.

20 is removed.

10 is removed.

10 is removed.

30 is removed.

30 is removed.

100 is not removed.

0 is not removed.

The last situation of the tree :

2 - 3

null

15 - 1

null

30 - 2

null

null

Running Command and Results-Q4

Some element has added to heap.

String representation of the heap :

10 - 3

40 - 3

30 - 2

20 - 2

50 - 1

60 - 1

70 - 1

The number of people younger than 15 : 3

The number of people younger than 5 : 0

The number of people younger than 80 : 13

The number of people younger than 40 : 7

The number of people younger than 45 : 10

The number of people older than 15 : 10

The number of people older than 80 : 0

The number of people older than 5 : 13

The number of people older than 15 : 3

The number of people older than 45 : 3

find the number of people at 30 : 30 - 2

find the number of people at 70 : 70 - 1

find the number of people at 100 : NullPointerException has been thrown because find method returned null.

7 is not removed.

50 is removed.

The heap after remove :

10 - 3

40 - 3

30 - 2

20 - 2

70 - 1

60 - 1

10 is removed.

The heap after remove :

40 - 3

10 - 2

30 - 2

20 - 2

70 - 1

60 - 1

10 is removed.

The heap after remove :

40 - 3

20 - 2

30 - 2

10 - 1

70 - 1

60 - 1

10 is removed.

The heap after remove :

40 - 3

20 - 2

30 - 2

60 - 1

70 - 1

10 is not removed.

The heap after remove :

40 - 3

20 - 2

30 - 2

60 - 1

70 - 1

40 is removed.

The heap after remove :

40 - 2

20 - 2

30 - 2

60 - 1

70 - 1

35 is not removed.

0 is not removed.

The heap after remove :

40 - 2

20 - 2

30 - 2

60 - 1

70 - 1