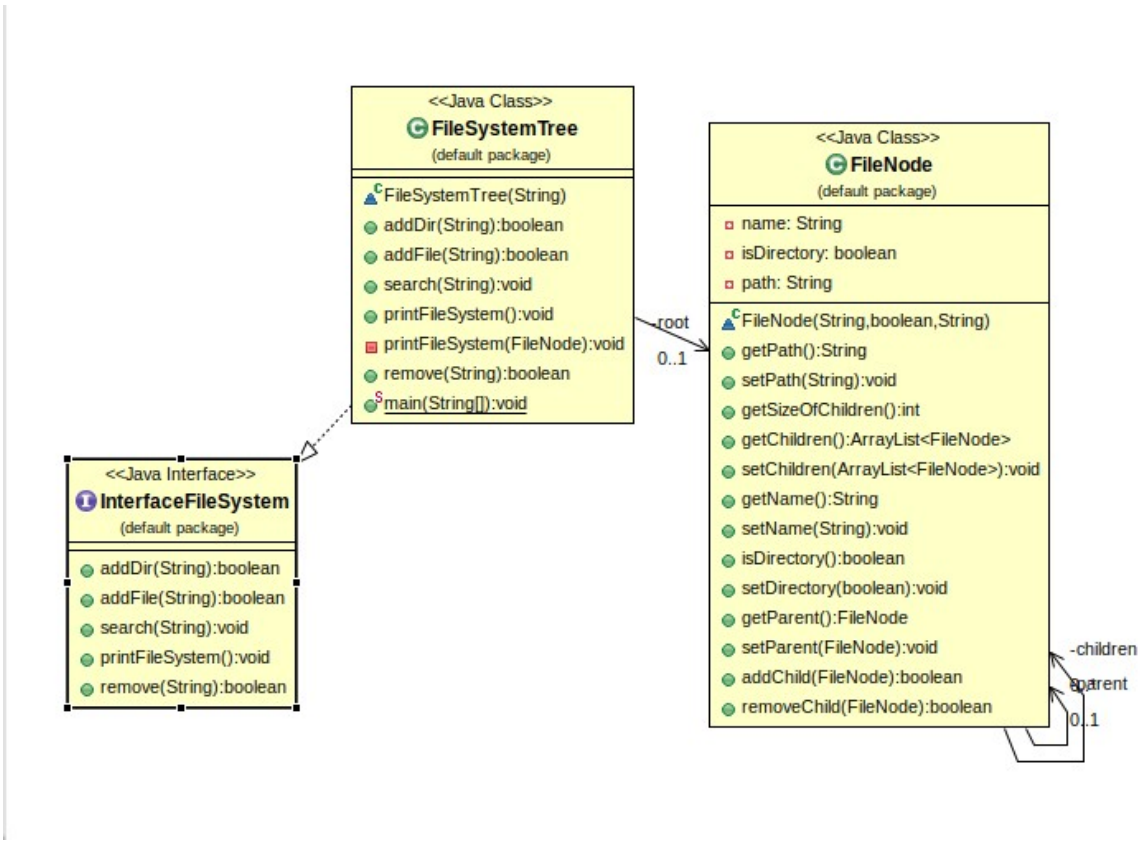
Report For HomeWork #5

Question 1)

Problem Solution Approach

In my FileSystemTree class,I thought that I need a File Node clas to handle tree system.Then I thought that a tree node has to use arraylist becouse of not knowing howmany children it has.Then I wrote methods which are addDir, addFile or remove etc. I used a boolean variable to discriminate file or directory.I also thought that I keep path of file becouseof printing problems.Becouse if search method calls,then it shows path of file. So I recod paths also as well as file names. I approach this problem like this.

Class Diagram

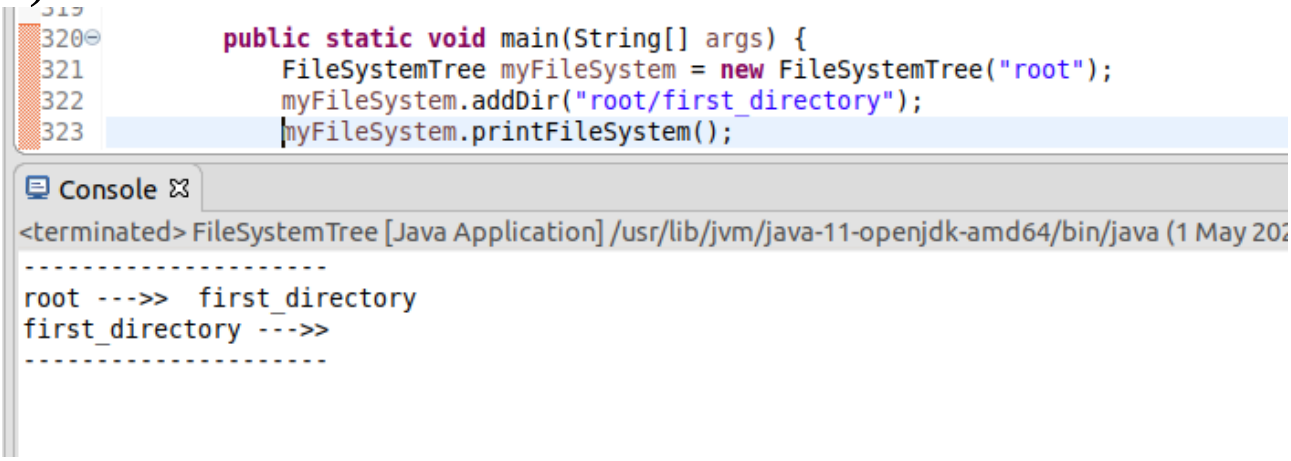


Test Cases

	Task	Parameter	Expected Result	Pass or not
1)	Add directory to root	Path	Should add	Pass
2)	Add file to root	Path	Should add	Pass
3)	Try add file to file	Path	Should not add	Pass
4)	Remove a file	Path	Should remove	Pass
5)	Remove directory	Path	Should ask and remove	Pass
6)	Remove root	Name of root	Should not remove	Pass
7)	Search a word	Word	Should list paths	Pass

Results of Test Cases

1)



2)

```
320        public static void main(String[] args) {
321            FileSystemTree myFileSystem = new FileSystemTree("root");
322            myFileSystem.addDir("root/first_directory");
323            myFileSystem.addFile("root/first_file.txt");
324            myFileSystem.printFileSystem();
```

Console

<terminated> FileSystemTree [Java Application] /usr/lib/jvm/java-11-openjdk-amd64/bin/java (1 Ma

```
--------------------
root --->>  first_directory  first_file.txt
first_directory --->>

--------------------
```

3)

```
320        public static void main(String[] args) {
321            FileSystemTree myFileSystem = new FileSystemTree("root");
322            myFileSystem.addDir("root/first_directory");
323            myFileSystem.addFile("root/first_directory/new_file.txt");
324            myFileSystem.addFile("root/first_file.txt");
325            System.out.println("I try to add file under file");
326            System.out.println(myFileSystem.addFile("root/first_file.txt/second.txt"));
327            myFileSystem.printFileSystem();
```

Console

<terminated> FileSystemTree [Java Application] /usr/lib/jvm/java-11-openjdk-amd64/bin/java (1 May 2020 02:03:52)

```
I try to add file under file
Your path is not valid becouse of file
false
--------------------
root --->>  first_directory  first_file.txt
first_directory --->>  new_file.txt


--------------------
```

4)

```
302        public static void main(String[] args) {
303            FileSystemTree myFileSystem = new FileSystemTree("root");
304            myFileSystem.addDir("root/first_directory");
305            myFileSystem.addFile("root/first_directory/new_file.txt");
306            myFileSystem.addFile("root/first_file.txt");
307            myFileSystem.printFileSystem();
308            System.out.println("After removing first_file.txt...");
309            myFileSystem.remove("root/first_file.txt");
310            myFileSystem.printFileSystem();
```

Console

<terminated> FileSystemTree [Java Application] /usr/lib/jvm/java-11-openjdk-amd64/bin/java (1 May 2020

```
----------------------
root --->>  first_directory  first_file.txt
first_directory --->>  new_file.txt


----------------------
After removing first_file.txt...
----------------------
root --->>  first_directory
first_directory --->>  new_file.txt

----------------------
```

5)

```
302        public static void main(String[] args) {
303            FileSystemTree myFileSystem = new FileSystemTree("root");
304            myFileSystem.addDir("root/first_directory");
305            myFileSystem.addFile("root/first_directory/new_file.txt");
306            myFileSystem.addFile("root/first_file.txt");
307            myFileSystem.printFileSystem();
308            System.out.println("After removing first_directory...");
309            myFileSystem.remove("root/first_directory");
310            myFileSystem.printFileSystem();
```

Console

<terminated> FileSystemTree [Java Application] /usr/lib/jvm/java-11-openjdk-amd64/bin/java (1 May 2020 02:11:21)

```
----------------------
root --->>  first_directory  first_file.txt
first_directory --->>  new_file.txt


----------------------
After removing first_directory...
"first_directory" Directory has these files or directories, Still Do you want to remove? Enter Yes or No
 new_file.txt
Yes
----------------------
root --->>  first_file.txt

----------------------
```

6)

```
302        public static void main(String[] args) {
303            FileSystemTree myFileSystem = new FileSystemTree("root");
304            myFileSystem.addDir("root/first_directory");
305            myFileSystem.addFile("root/first_directory/new_file.txt");
306            myFileSystem.addFile("root/first_file.txt");
307            myFileSystem.printFileSystem();
308            System.out.println(myFileSystem.remove("root"));
309
310            myFileSystem.printFileSystem();
```

Console

<terminated> FileSystemTree [Java Application] /usr/lib/jvm/java-11-openjdk-amd64/bin/java (1 May 2020

```
---------------------
root --->>  first_directory  first_file.txt
first_directory --->>  new_file.txt


---------------------
false
---------------------
root --->>  first_directory  first_file.txt
first_directory --->>  new_file.txt


---------------------
```

7)

```
301
302        public static void main(String[] args) {
303            FileSystemTree myFileSystem = new FileSystemTree("root");
304            myFileSystem.addDir("root/first_directory");
305            myFileSystem.addFile("root/first_directory/new_file.txt");
306            myFileSystem.addDir("root/second_directory");
307            myFileSystem.addDir("root/second_directory/new_directory");
308            myFileSystem.addFile("root/second_directory/new_directory/new_file.doc");
309            myFileSystem.addDir("root/second_directory/new_directory2");
310            myFileSystem.addFile("root/second_directory/new_directory/new_file2.doc");
311            myFileSystem.printFileSystem();
312            myFileSystem.search("new");
```

Console

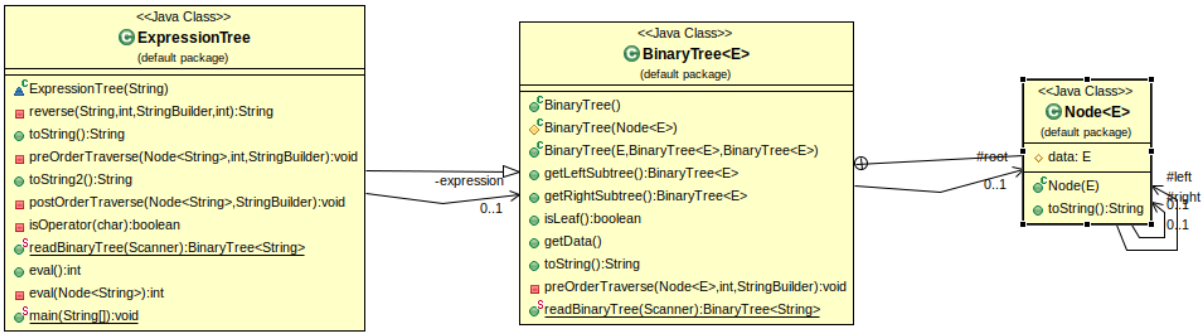
```
<terminated> FileSystemTree [Java Application] /usr/lib/jvm/java-11-openjdk-amd64/bin/java (1 May 2020 02:14:53)
--------------------
root --->>  first_directory  second_directory
first_directory --->>  new_file.txt

second_directory --->>  new_directory  new_directory2
new_directory --->>  new_file.doc  new_file2.doc


new_directory2 --->>
--------------------
file - root/first_directory/new_file.txt
dir - root/second_directory/new_directory
dir - root/second_directory/new_directory2
file - root/second_directory/new_directory/new_file.doc
file - root/second_directory/new_directory/new_file2.doc
```

Question 2)

Problem Solution Approach

In my ExpressionTree class,We are asked for a binary tree that reads both prefix and postfix expressions.I thought like that,When I take postfix expression , I turn this postfix expression to prefix expression , then send to the read method. So If I write readbinarytree method for prefix , I also fix for postfix expressions. When I turn the postfix expression to prefix expression , I used reverse method that implemented by me.So the tree is constructed by sending prefix or postfix expressions.After the tree is constructed, other methods will be recursively.I approach this problem so close to this.

Class Diagram



Test Cases

	Task	Parameter	Expected Result	Pass or not
1)	Send prefix and use toSting method	Prefix expression	Should show prefix version	Pass
2)	Send prefix and use toString2 method	Prefix expression	Shoud show postfix version	Pass
3)	Send postfix and use toSting method	Postfix expression	Should show prefix version	Pass
4)	Send postfix and use toString 2 method	Postfix expression	Should show postfix version	Pass
5)	Use eval method	-	Should evaluate expression tree	Pass

Results of Test Cases

1)

```
182        public static void main(String args[]) {
183            String input="+ + 10 * 5 15 20";
184            ExpressionTree obj=new ExpressionTree(input);
185            System.out.println(obj.toString());
```

Console

<terminated> ExpressionTree [Java Application] /usr/lib/jvm/java-11-openjdk-amd64/bin/j

```
+ + 10 * 5 15 20
```

2)

```
182        public static void main(String args[]) {
183            String input="+ + 10 * 5 15 20";
184            ExpressionTree obj=new ExpressionTree(input);
185            System.out.println("prefix version = ");
186            System.out.println(obj.toString());
187            System.out.println("postfix version = ");
188            System.out.println(obj.toString2());
```

Console

<terminated> ExpressionTree [Java Application] /usr/lib/jvm/java-11-openjdk-amd64

```
prefix version =
 + + 10 * 5 15 20
postfix version =
10 5 15 * + 20 +
```

3)

```
181
182        public static void main(String args[]) {
183            String input="10 5 15 * + 20 +";
184            ExpressionTree obj=new ExpressionTree(input);
185            System.out.println("prefix version = ");
186            System.out.println(obj.toString());
```

Console

```
<terminated> ExpressionTree [Java Application] /usr/lib/jvm/java-11-openjdk-amd64/b
prefix version =
 + 20 + * 15 5 10
```

4)

```
182        public static void main(String args[]) {
183            String input="10 5 15 * + 20 +";
184            ExpressionTree obj=new ExpressionTree(input);
185            System.out.println("prefix version = ");
186            System.out.println(obj.toString());
187            System.out.println("postfix version = ");
188            System.out.println(obj.toString2());
```

Console

```
<terminated> ExpressionTree [Java Application] /usr/lib/jvm/java-11-openjdk-am
prefix version =
 + 20 + * 15 5 10
postfix version =
20 15 5 * 10 + +
```

5)

```
182        public static void main(String args[]) {
183            String input1="+ + 10 * 5 15 20";
184            System.out.println("Expression is : "+ input1);
185            ExpressionTree obj1=new ExpressionTree(input1);
186            System.out.println("prefix version = ");
187            System.out.println(obj1.toString());
188            System.out.println("postfix version = ");
189            System.out.println(obj1.toString2());
190            System.out.println("Evaluation is : "+ obj1.eval());
191            System.out.println();
192            String input="10 5 15 * + 20 +";
193            System.out.println("Expression is : "+ input);
194            ExpressionTree obj2=new ExpressionTree(input);
195            System.out.println("prefix version = ");
196            System.out.println(obj2.toString());
197            System.out.println("postfix version = ");
```

Console

<terminated> ExpressionTree [Java Application] /usr/lib/jvm/java-11-openjdk-amd64/bin/java

```
Expression is : + + 10 * 5 15 20
prefix version =
 + + 10 * 5 15 20
postfix version =
10 5 15 * + 20 +
Evaluation is : 105

Expression is : 10 5 15 * + 20 +
prefix version =
 + 20 + * 15 5 10
postfix version =
20 15 5 * 10 + +
Evaluation is : 105
```

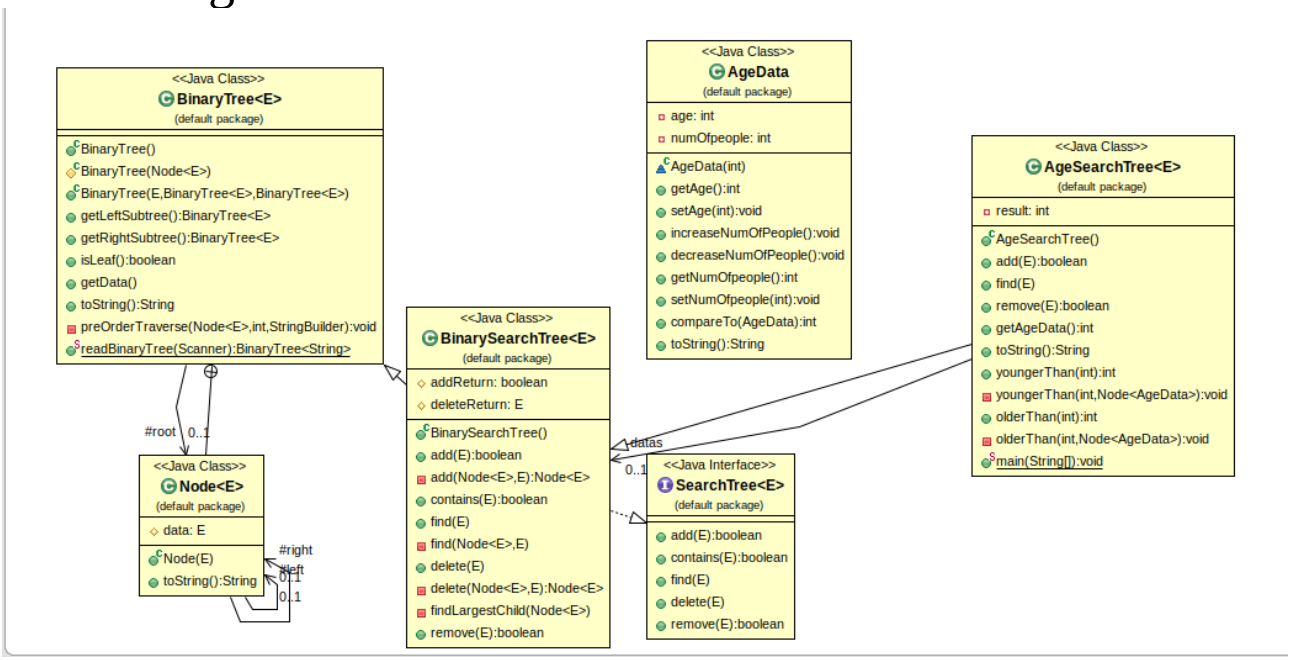
Question 3)

Problem Solution Approach

In my AgeSearchTree class,Firstly I wrote AgeData class to handle AgeSearchTree class. In AgeData class I wrote compareTo method to compare objects. So AgeData Class is a Comparable class.Then I wrote AgeSearchTree class as a generic class but it behaves like concrete class

becouse of AgeData. I wrote youngerThan and olderThan methods recursively and it doesn’t traverse every node to find youngers and olders.It traverse if statement is young and old. It uses whatever it needs and chooses that subtree then traverse. I approach this problem very close to this.

Class Diagram



Test Cases

	Task	Parameter	Expected Result	Pass or not
1)	Add some age same and different	AgeData	Sould add	Pass
2)	Use remove method	AgeData	Should remove	Pass
3)	Use find method	AgeData	Should find	Pass
4)	Use youngerThan method	Int age	Should return younger number of people	Pass
5)	Use olderThan method	Int age	Should return older number of people	Pass

Results of Test Cases

1)

```
148    public static void main(String args[]) {
149        AgeSearchTree<AgeData> ageTree = new AgeSearchTree<AgeData>();
150        ageTree.add(new AgeData(10));
151        ageTree.add(new AgeData(20));
152        ageTree.add(new AgeData(5));
153        ageTree.add(new AgeData(10));
154        ageTree.add(new AgeData(15));
155
156        System.out.println(ageTree.toString());
```

Console

<terminated> AgeSearchTree [Java Application] /usr/lib/jvm/java-11-openjdk-amd64/bin/java (1 May 20

```
10 - 2
 5 - 1
  null
  null
 20 - 1
  15 - 1
   null
   null
  null
```

2)

```
148    public static void main(String args[]) {
149        AgeSearchTree<AgeData> ageTree = new AgeSearchTree<AgeData>();
150        ageTree.add(new AgeData(10));
151        ageTree.add(new AgeData(20));
152        ageTree.add(new AgeData(5));
153        ageTree.add(new AgeData(10));
154        ageTree.add(new AgeData(15));
155
156        System.out.println(ageTree.toString());
157        ageTree.remove(new AgeData(10));
158        ageTree.remove(new AgeData(15));
159        System.out.println("After removing AgeData(10) and AgeData(15)...");
160        System.out.println(ageTree.toString());
```

Console

<terminated> AgeSearchTree [Java Application] /usr/lib/jvm/java-11-openjdk-amd64/bin/java (1 May 2020 02:40:18)

```
10 - 2
 5 - 1
  null
  null
 20 - 1
  15 - 1
   null
   null
  null

After removing AgeData(10) and AgeData(15)...
10 - 1
 5 - 1
  null
  null
 20 - 1
  null
  null
```

3)

```
148    public static void main(String args[]) {
149        AgeSearchTree<AgeData> ageTree = new AgeSearchTree<AgeData>();
150        ageTree.add(new AgeData(10));
151        ageTree.add(new AgeData(20));
152        ageTree.add(new AgeData(5));
153        ageTree.add(new AgeData(10));
154        ageTree.add(new AgeData(15));
155
156        System.out.println(ageTree.toString());
157        System.out.println(ageTree.find(new AgeData(20)).toString());
```

Console

<terminated> AgeSearchTree [Java Application] /usr/lib/jvm/java-11-openjdk-amd64/bin/java (1 May 202

```
10 - 2
 5 - 1
  null
  null
 20 - 1
  15 - 1
   null
   null
  null

20 - 1
```

4)

```
148    public static void main(String args[]) {
149        AgeSearchTree<AgeData> ageTree = new AgeSearchTree<AgeData>();
150        ageTree.add(new AgeData(10));
151        ageTree.add(new AgeData(20));
152        ageTree.add(new AgeData(5));
153        ageTree.add(new AgeData(10));
154        ageTree.add(new AgeData(15));
155
156        System.out.println(ageTree.toString());
157
158        System.out.println("Number of young people than 15");
159        System.out.println(ageTree.youngerThan(15));
```

Console

<terminated> AgeSearchTree [Java Application] /usr/lib/jvm/java-11-openjdk-amd64/bin/java (1 May 2020

```
10 - 2
 5 - 1
  null
  null
 20 - 1
  15 - 1
   null
   null
  null

Number of young people than 15
3
```

5)

```
148    public static void main(String args[]) {
149        AgeSearchTree<AgeData> ageTree = new AgeSearchTree<AgeData>();
150        ageTree.add(new AgeData(10));
151        ageTree.add(new AgeData(20));
152        ageTree.add(new AgeData(5));
153        ageTree.add(new AgeData(10));
154        ageTree.add(new AgeData(15));
155
156        System.out.println(ageTree.toString());
157
158        System.out.println("Number of old people than 10");
159        System.out.println(ageTree.olderThan(10));
```

Console

<terminated> AgeSearchTree [Java Application] /usr/lib/jvm/java-11-openjdk-amd64/bin/java (1

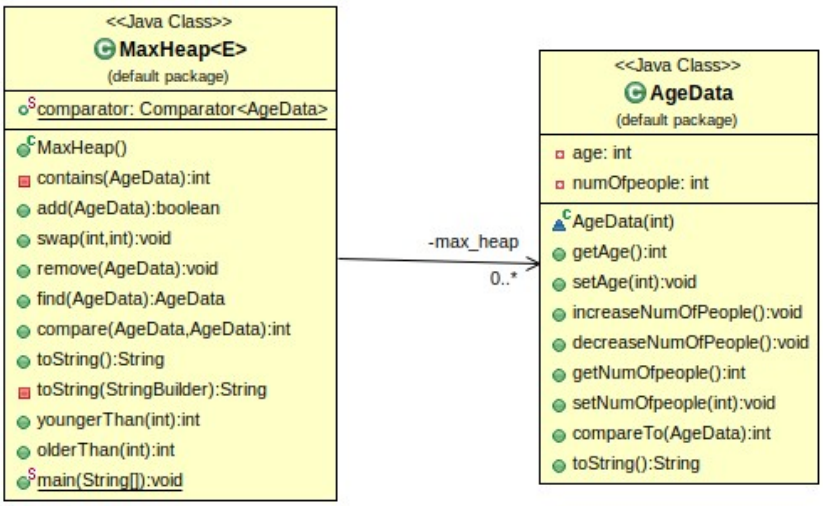
```
10 - 2
 5 - 1
  null
  null
 20 - 1
  15 - 1
   null
   null
  null

Number of old people than 10
2
```

Question 4)

Problem Solution Approach

In my MaxHeap class, I thought like AgeSearchTree class but I used heap algorithms to handle this class. And also youngerThan and olderThan methods have to traverse every node becouse the key is not age but number of people. And I wrote a comparator method to compare AgeData class and also I changed compareTo method of AgeData Class becouse of key is number of people. I approach this problem like this.

Class Diagram



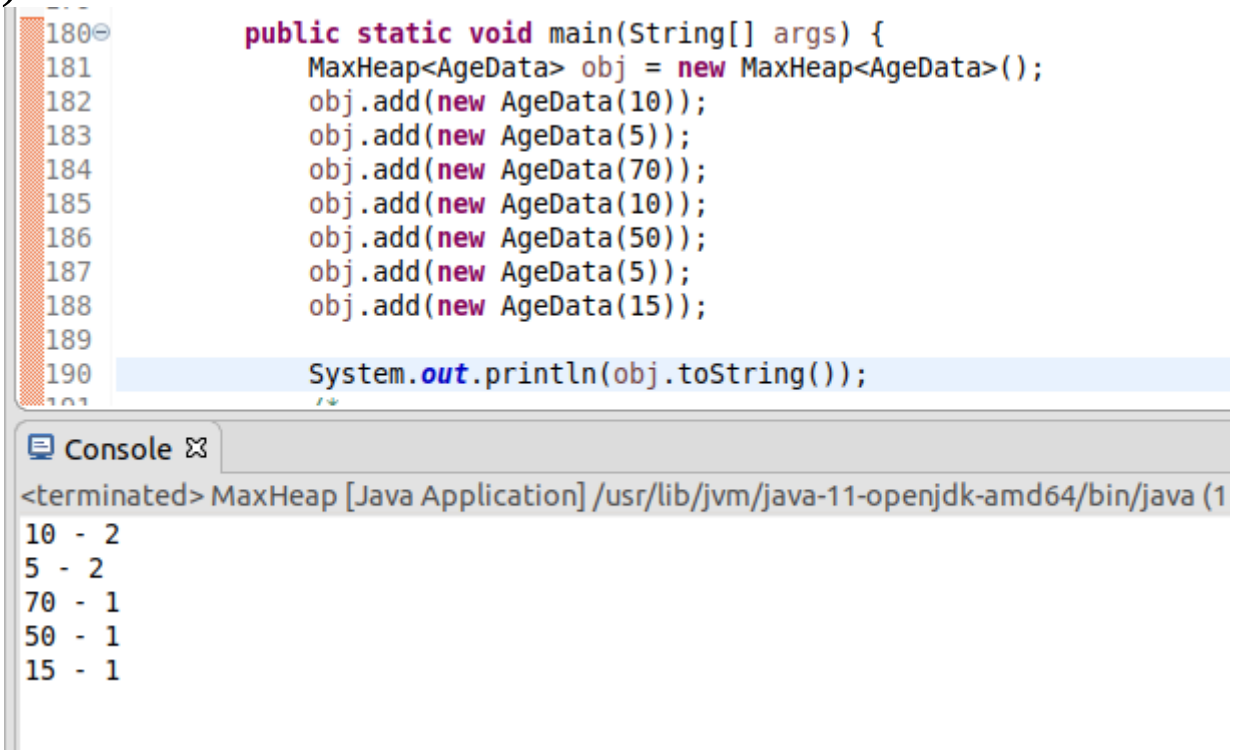
Test Cases

	Task	Parameter	Expected Result	Pass or not
1)	Add some age same and different	AgeData	Sould add	Pass
2)	Use remove method	AgeData	Should remove	Pass
3)	Use find method	AgeData	Should find	Pass
4)	Use youngerThan	Int age	Should return	Pass

	method		younger number of people	
5)	Use olderThan method	Int age	Should return older number of people	Pass

Results of Test Cases

1)



2)

```
180        public static void main(String[] args) {
181            MaxHeap<AgeData> obj = new MaxHeap<AgeData>();
182            obj.add(new AgeData(10));
183            obj.add(new AgeData(5));
184            obj.add(new AgeData(70));
185            obj.add(new AgeData(10));
186            obj.add(new AgeData(50));
187            obj.add(new AgeData(5));
188            obj.add(new AgeData(15));
189
190            System.out.println(obj.toString());
191            obj.remove(new AgeData(10));
192            System.out.println("After removing 10");
193            String heapstr=obj.toString();
194            System.out.println(heapstr);
```

Console

<terminated> MaxHeap [Java Application] /usr/lib/jvm/java-11-openjdk-amd64/bin/java (1

```
10 - 2
5 - 2
70 - 1
50 - 1
15 - 1

After removing 10
5 - 2
10 - 1
70 - 1
50 - 1
15 - 1
```

3)

```
180        public static void main(String[] args) {
181            MaxHeap<AgeData> obj = new MaxHeap<AgeData>();
182            obj.add(new AgeData(10));
183            obj.add(new AgeData(5));
184            obj.add(new AgeData(70));
185            obj.add(new AgeData(10));
186            obj.add(new AgeData(50));
187            obj.add(new AgeData(5));
188            obj.add(new AgeData(15));
189
190            System.out.println(obj.toString());
191
192            System.out.println(obj.find(new AgeData(10)).toString());
```

Console

<terminated> MaxHeap [Java Application] /usr/lib/jvm/java-11-openjdk-amd64/bin/java (1 May 2020

```
10 - 2
5 - 2
70 - 1
50 - 1
15 - 1

10 - 2
```

4)

```
180        public static void main(String[] args) {
181            MaxHeap<AgeData> obj = new MaxHeap<AgeData>();
182            obj.add(new AgeData(10));
183            obj.add(new AgeData(5));
184            obj.add(new AgeData(70));
185            obj.add(new AgeData(10));
186            obj.add(new AgeData(50));
187            obj.add(new AgeData(5));
188            obj.add(new AgeData(15));
189
190            System.out.println(obj.toString());
191            System.out.println("Number of people that is younger than 13");
192            System.out.println(obj.youngerThan(13));
193
```

Console

<terminated> MaxHeap [Java Application] /usr/lib/jvm/java-11-openjdk-amd64/bin/java (1 May 2020 02:

```
10 - 2
5 - 2
70 - 1
50 - 1
15 - 1

Number of people that is younger than 13
4
```

5)

```
180⊖        public static void main(String[] args) {
181             MaxHeap<AgeData> obj = new MaxHeap<AgeData>();
182             obj.add(new AgeData(10));
183             obj.add(new AgeData(5));
184             obj.add(new AgeData(70));
185             obj.add(new AgeData(10));
186             obj.add(new AgeData(50));
187             obj.add(new AgeData(5));
188             obj.add(new AgeData(15));
189
190             System.out.println(obj.toString());
191             System.out.println("Number of people that is older than 10");
192             System.out.println(obj.olderThan(10));
```

Console

<terminated> MaxHeap [Java Application] /usr/lib/jvm/java-11-openjdk-amd64/bin/java (1 May 2020 02

```
10 - 2
5 - 2
70 - 1
50 - 1
15 - 1

Number of people that is older than 10
3
```

Mustafa Tokgöz
171044077