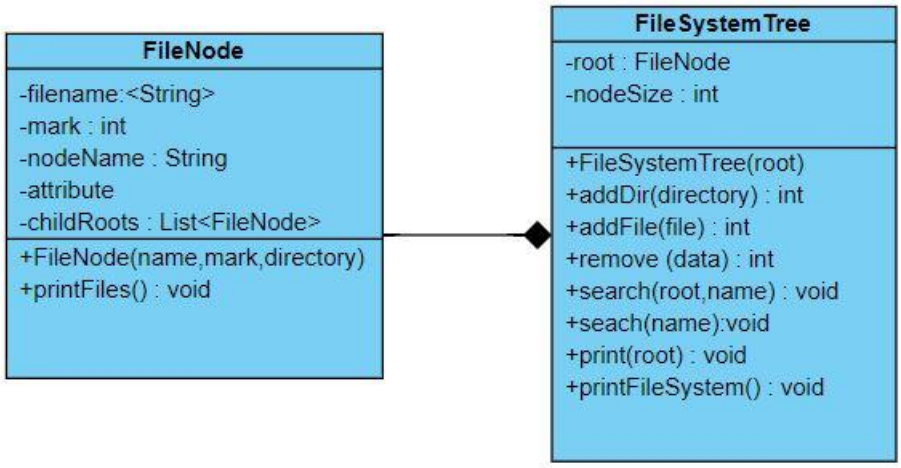
GIT Department of Computer Engineering
CSE 222/505 - Spring 2020
Homework 5 Report

Oğuzhan SEZGİN
1801042005

Q1

CLASS DIAGRAM



Program works as a desired. It add file or directory and remove it and search it.There are some test cases below.

PROBLEM SOLUTION APPROACH

If user wants to remove root of tree program warns user .
If tree is empty and user wants to remove some directories or files, program warns user.
If user wants to remove not exist file or not exist directory program warns user.
If user wants to add file or directory to a file program warns user.

TEST CASES

Program adds files and directories correctly.If user wants to remove a directory which has some other directories and files ask user to sure remove.

```
8         FileSystemTree fs=new FileSystemTree("root");
9         //------------------ ADD-REMOVE TEST ----------------------------
10        fs.addDir("root/dir1");
11        fs.addFile("root/dir1/F1.txt");
12        fs.addDir("root/dir1/dir2");
13        fs.addFile("root/dir1/dir2/F2.txt");
14        fs.addFile("root/dir1/dir2/F3.txt");
15        fs.addFile("root/dir1/dir2/F4.txt");
16        fs.addDir("root/dir1/dir2/dir7");
17        fs.addDir("root/dir1/dir3");
18        fs.addFile("root/dir1/dir3/F5.txt");
19        fs.addDir("root/dir1/dir3/dir4");
20        fs.addDir("root/dir1/dir3/dir4/F6");
21        fs.addDir("root/dir1/dir5");
22        fs.addDir("root/dir6");
23        System.out.println("-------Before remove-------------");
24        fs.printFileSystem();
25        System.out.println("");
26        fs.remove("root/dir1/dir3");
27        System.out.println("-------After remove-------------");
28        fs.printFileSystem();
29
30
31
32
```

```
Console  Debug
<terminated> main [Java Application] C:\Program Files\Java\jre1.8.0_251\bin\javaw.exe (3 May 2020 11:52:23)
-------Before remove-------------
root(d) dir1(d) F1.txt(f) dir2(d) F2.txt(f) F3.txt(f) F4.txt(f) dir7(d) dir3(d) F5.txt(f) dir4(d) F6(d) dir5(d) dir6(d)

" root/dir1/dir3 " directory contains below elements.Do you want to removethis directory ?(Y/N)

dir3(d) F5.txt(f) dir4(d) F6(d) y
-------After remove-------------
root(d) dir1(d) F1.txt(f) dir2(d) F2.txt(f) F3.txt(f) F4.txt(f) dir7(d) dir5(d) dir6(d)
```

If user wants to search some directories or files program prints path of directory or file

```
36          //-------------------SEARCH TEST --------------------
37          fs.addDir("root/dir1");
38          fs.addFile("root/dir1/F1.txt");
39          fs.addDir("root/dir1/dir2");
40          fs.addFile("root/dir1/dir2/F2.txt");
41          fs.addFile("root/dir1/dir2/F3.txt");
42          fs.addFile("root/dir1/dir2/F4.txt");
43          fs.addDir("root/dir1/dir2/dir7");
44          fs.addDir("root/dir1/dir3");
45          fs.addFile("root/dir1/dir3/F5.txt");
46          fs.addDir("root/dir1/dir3/dir4");
47          fs.addDir("root/dir1/dir3/dir4/F6");
48          fs.addDir("root/dir1/dir5");
49          fs.addDir("root/dir6");
50          fs.addFile("root/dir6/F2.txt");
51          System.out.println("-------------F2.txt search-------------");
52          fs.search( "F2.txt");
53          System.out.println("-------------dir2.txt search-------------");
54          fs.search( "dir2");
55
56
```

Console Debug

<terminated> main [Java Application] C:\Program Files\Java\jre1.8.0_251\bin\javaw.exe (3 May 2020 11:57:42)

```
-------------F2.txt search-------------
file - root/dir1/dir2
file - root/dir6
-------------dir2.txt search-------------
dir - root/dir1/dir2
```

There are some exception testing below

```
58          //--------------------- EXCEPTION TEST -----------------------
59          System.out.println("-----------Empty tree remove test-----------");
60          fs.remove("d3");
61          fs.addDir("root/dir1");
62          fs.addFile("root/dir1/F1.txt");
63          fs.addDir("root/dir1/dir2");
64          fs.addFile("root/dir1/dir2/F2.txt");
65          System.out.println("-----------F3.txt remove test-----------");
66          fs.remove("F3.txt");
67          System.out.println("-----------Root remove test-----------");
68          fs.remove("root");
69          System.out.println("-----------Add file to directory test-----------");
70          fs.addDir("root/dir1/F1.txt/F4.txt");
71
72      }
```

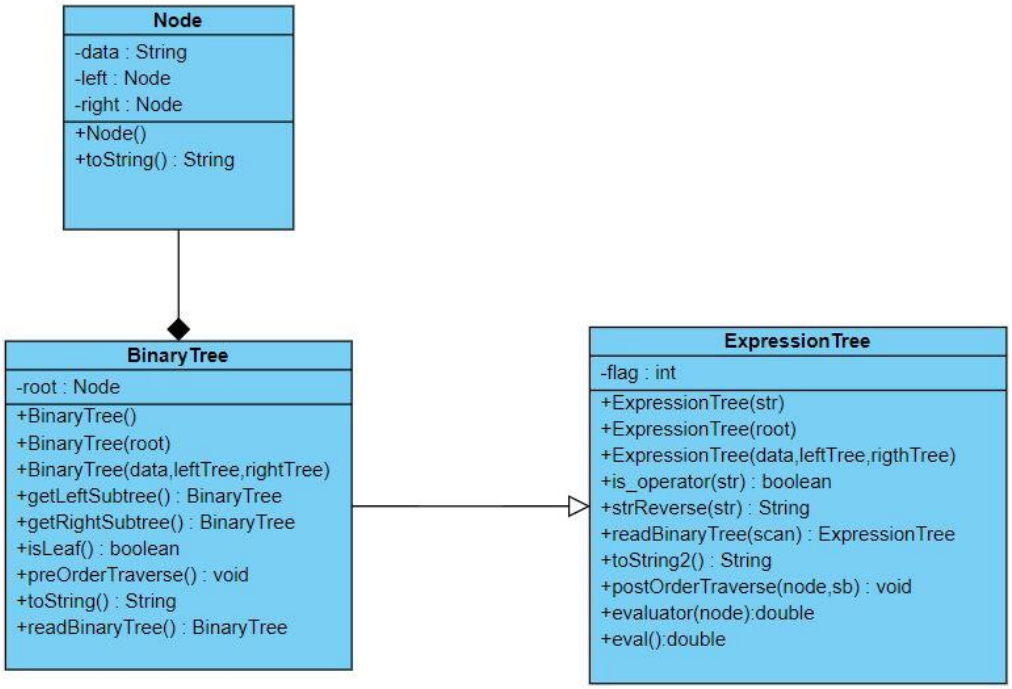
Console Debug

<terminated> main [Java Application] C:\Program Files\Java\jre1.8.0_251\bin\javaw.exe (3 May 2020 12:08:18)

```
-----------Empty tree remove test-----------
There is no directory or file!!!
-----------F3.txt remove test-----------
There is no this name directory or file
-----------Root remove test-----------
Root can not remove!!!
-----------Add file to directory test-----------
Wrong Directory !!!
```

Q2

CLASS DIAGRAM



Program works as a desired.It reads expression with using overridden readBinaryTree method.It can reads both postfix or prefix expression and construct binary tree.It can evaluate expression and prints result.It can traverse post order or pre order traverse method.There are some test cases below.

PROBLEM SOLUTION APPROACH

If user enters missing number program warns user.
If user enters non numerical character program warns user.

TEST CASES

Program evaluates post or pre order expression.It print tree with toString and toString2 method.

```
11              //----------------------EVALUATION TEST ---------------------
12              ExpressionTree expt = new ExpressionTree("+ + 10 * 5 15 20");
13              ExpressionTree expt2 = new ExpressionTree("+ - 100 / 5 15 20");
14              ExpressionTree expt3 = new ExpressionTree("10 2 8 * + 3 - ");
15              ExpressionTree expt4 = new ExpressionTree("4 55 + 62 23 - / ");
16              System.out.println("-------------Evaluation -------------");
17              System.out.println(expt.eval());
18              System.out.println(expt2.eval());
19              System.out.println(expt3.eval());
20              System.out.println(expt4.eval());
21              System.out.println("-------------toString2 -------------");
22              System.out.println(expt.toString2());
23              System.out.println(expt2.toString2());
24              System.out.println(expt3.toString2());
25              System.out.println(expt4.toString2());
26              System.out.println("-------------toString -------------");
27              System.out.println(expt.toString());
28              System.out.println(expt2.toString());
29              System.out.println(expt3.toString());
30              System.out.println(expt4.toString());
31          }
32          catch (InputMismatchException e) {
```

Console Debug

<terminated> main (1) [Java Application] C:\Program Files\Java\jre1.8.0_251\bin\javaw.exe (3 May 2020 12:52:29)

```
105.0
119.66666666666667
23.0
1.5128205128205128
-------------toString2 -------------
10 5 15 * + 20 +
100 5 15 / - 20 +
10 2 8 * + 3 -
4 55 + 62 23 - /
-------------toString -------------
+ + 10 * 5 15 20
+ - 100 / 5 15 20
- + 10 * 2 8 3
/ + 4 55 - 62 23
```

There are some exception tests.

```
33          //----------------EXCEPTİON TEST ---------------------
34          ExpressionTree expt = new ExpressionTree("+ + 10 * 5  20");
35          System.out.println(expt.eval());
36
37
```

Console Debug

```
<terminated> main (1) [Java Application] C:\Program Files\Java\jre1.8.0_251\bin\javaw.exe (3 May 2020 12:56:17)
Missing char  !!!
```

```
33          //----------------EXCEPTION TEST ---------------------
34          ExpressionTree expt = new ExpressionTree("+ + 10 * 5 15 a 20");
35          System.out.println(expt.eval());
36
37
```

Console Debug

```
<terminated> main (1) [Java Application] C:\Program Files\Java\jre1.8.0_251\bin\javaw.exe (3 May 2020 12:57:36)
Invalid input !!!
```

```
33          //----------------EXCEPTİON TEST ---------------------
34          ExpressionTree expt = new ExpressionTree("+ + 10 * 5 15a 20");
35          System.out.println(expt.eval());
36
37
```

Console Debug

```
<terminated> main (1) [Java Application] C:\Program Files\Java\jre1.8.0_251\bin\javaw.exe (3 May 2020 12:58:07)
Invalid input !!!
```

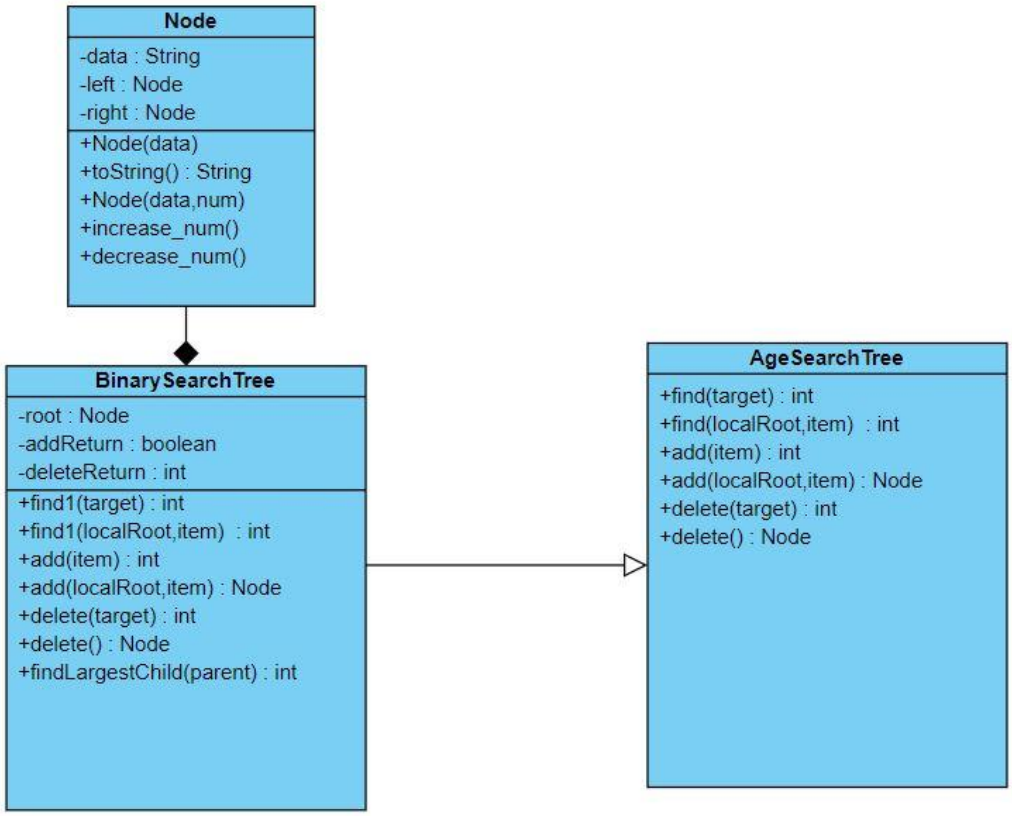
```
34          ExpressionTree expt = new ExpressionTree("+ + 10 * 5  20");
35          System.out.println(expt.toString2());
36
37
```

Console Debug

```
<terminated> main (1) [Java Application] C:\Program Files\Java\jre1.8.0_251\bin\javaw.exe (3 May 2020 12:55:14)
Missing char  !!!
```

Q3

CLASS DIAGRAM



Program works as a desired.It can add age and if age to add is exist in tree then program increase age number but if there is not exist it add new node to tree.It can remove age and if age to remove is exist more than 1 in tree then program decrease age number but if there is exist 1 it remove node to tree.It can find age and prints number of age.It can print younger than or older ages than given age.There is some test cases below.

PROBLEM SOLUTION APPROACH

If user wants to remove not exist number program warn user.
If user wants to prints not exist node with toString method ,program warn user.

TEST CASES

Program add and prints number correctly.

```
10
11              //-----------------ADD --------------------
12              AgeSearchTree age = new AgeSearchTree();
13              age.add(new AgeData(10));
14              age.add(new AgeData(21));
15              age.add(new AgeData(20));
16              age.add(new AgeData(16));
17              age.add(new AgeData(5));
18              age.add(new AgeData(11));
19              age.add(new AgeData(12));
20              age.add(new AgeData(12));
21              age.add(new AgeData(10));
22
23              System.out.println(age.toString());
24
25
```

Console

<terminated> Main [Java Application] C:\Program Files\Java\jre1.8.0_251\bin\javaw.exe (3 May 2020

```
10-2
5-1
null
null
21-1
20-1
16-1
11-1
null
12-2
null
null
null
null
null
```

Program removes correctly.

```
9          try {
10
11             //----------------ADD -------------------
12             AgeSearchTree age = new AgeSearchTree();
13             age.add(new AgeData(10));
14             age.add(new AgeData(21));
15             age.add(new AgeData(20));
16             age.add(new AgeData(16));
17             age.add(new AgeData(5));
18             age.add(new AgeData(11));
19             age.add(new AgeData(12));
20             age.add(new AgeData(12));
21             age.add(new AgeData(10));
22             age.remove(12);
23             age.remove(21);
24
25             System.out.println(age.toString());
26
```

Console

<terminated> Main [Java Application] C:\Program Files\Java\jre1.8.0_251\bin\javaw.exe (3 M

```
10-2
5-1
null
null
20-1
16-1
11-1
null
12-1
null
null
null
null
```

Program finds correctly.

```
10
11              //----------------ADD -------------------
12              AgeSearchTree age = new AgeSearchTree();
13              age.add(new AgeData(10));
14              age.add(new AgeData(21));
15              age.add(new AgeData(20));
16              age.add(new AgeData(16));
17              age.add(new AgeData(5));
18              age.add(new AgeData(11));
19              age.add(new AgeData(12));
20              age.add(new AgeData(12));
21              age.add(new AgeData(10));
22
23              //----------------FIND TEST -------------------
24              System.out.println("Find and tostring 12");
25              System.out.println(age.find(new AgeData(12)));
26              System.out.println("Find and tostring 16");
27              System.out.println(age.find(new AgeData(16)));
28
29
```

Console

<terminated> Main [Java Application] C:\Program Files\Java\jre1.8.0_251\bin\javaw.exe (3 May 20

```
Find and tostring 12
12-2
Find and tostring 16
16-1
```

Program prints correctly younger and older ages.

```
11          //----------------ADD -------------------
12          AgeSearchTree age = new AgeSearchTree();
13          age.add(new AgeData(10));
14          age.add(new AgeData(21));
15          age.add(new AgeData(20));
16          age.add(new AgeData(16));
17          age.add(new AgeData(5));
18          age.add(new AgeData(11));
19          age.add(new AgeData(12));
20          age.add(new AgeData(12));
21          age.add(new AgeData(10));
22          //--------------YOUNGER AND OLDER THAN TEST--------------------
23
24          System.out.println("----------- Younger than 10 ------------------");
25          System.out.println(age.youngerThan(10));
26          System.out.println("----------- Older than 10 ------------------");
27          System.out.println(age.olderThan(10));
28
```

Console

<terminated> Main [Java Application] C:\Program Files\Java\jre1.8.0_251\bin\javaw.exe (3 May 2020 16:34:08)

```
----------- Younger than 10 ------------------
1
----------- Older than 10 ------------------
6
```

There is some exception test below.

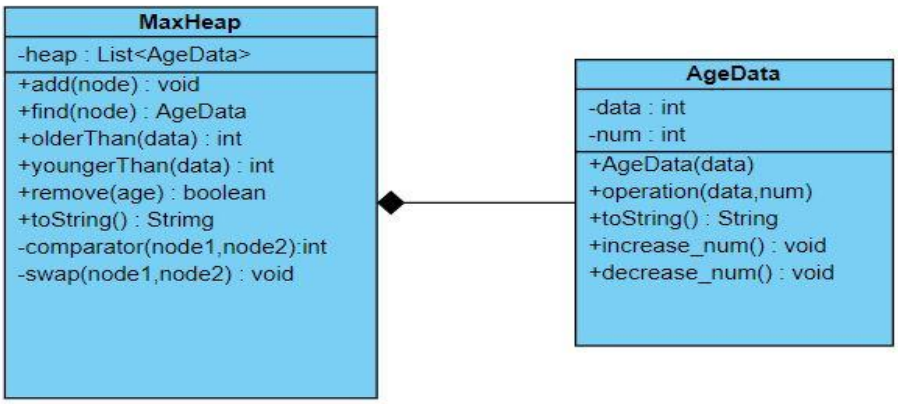
```
10
11              //-----------------ADD --------------------
12              AgeSearchTree age = new AgeSearchTree();
13              age.add(new AgeData(10));
14              age.add(new AgeData(21));
15              age.add(new AgeData(20));
16              age.add(new AgeData(16));
17              age.add(new AgeData(5));
18              age.add(new AgeData(11));
19              age.add(new AgeData(12));
20              age.add(new AgeData(12));
21              age.add(new AgeData(10));
22
23              //----------------EXCEPTION TEST -------------------
24              System.out.println("Remove not exist age");
25              age.remove(53);
26              System.out.println("Find not exist age");
27              System.out.println(age.find(new AgeData(40)).toString());
28
```

Console

```
<terminated> Main [Java Application] C:\Program Files\Java\jre1.8.0_251\bin\javaw.exe (3 May 2020 16:38
Remove not exist age
Program can not delete.There is no this number age
Find not exist age
There is no this number age
```

Q3

CLASS DIAGRAM



Program works as a desired.Heap sorted with binary tree rules which writes in the book. Program adds age and remove ages and find ages.Also it prints older and younger age number.There are test cases below.

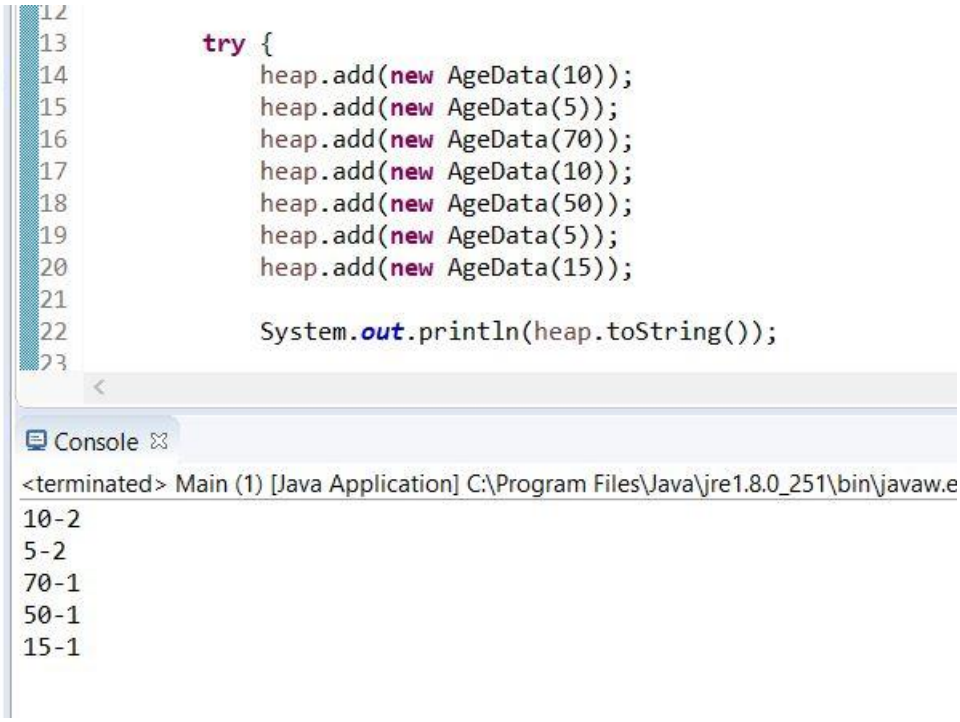
PROBLEM SOLUTION APPROACH

If user wants to remove not exist number program warn user.

If user wants to prints not exist node with toString method ,program warn user.

TEST CASES

Program add and prints number correctly.



Program removes correctly.

```
14          heap.add(new AgeData(10));
15          heap.add(new AgeData(5));
16          heap.add(new AgeData(70));
17          heap.add(new AgeData(10));
18          heap.add(new AgeData(50));
19          heap.add(new AgeData(5));
20          heap.add(new AgeData(15));
21          heap.remove(10);
22          heap.remove(70);
23          System.out.println(heap.toString());
24
```

Console

```
<terminated> Main (1) [Java Application] C:\Program Files\Java\jre1.8.0_251\bi
10-1
5-2
15-1
50-1
```

Program prints correctly younger and older ages.

```
13      try {
14          heap.add(new AgeData(10));
15          heap.add(new AgeData(5));
16          heap.add(new AgeData(70));
17          heap.add(new AgeData(10));
18          heap.add(new AgeData(50));
19          heap.add(new AgeData(5));
20          heap.add(new AgeData(15));
21          //---------------YOUNGER - OLDER AGE TEST --------------------
22          System.out.println(heap.youngerThan(16));
23          System.out.println(heap.olderThan(10));
24
```

Console

```
<terminated> Main (1) [Java Application] C:\Program Files\Java\jre1.8.0_251\bin\javaw.exe (3 May 2020 18
5
3
```

Program finds correctly.

```
11          MaxHeap heap=new MaxHeap();
12
13          try {
14              heap.add(new AgeData(10));
15              heap.add(new AgeData(5));
16              heap.add(new AgeData(70));
17              heap.add(new AgeData(10));
18              heap.add(new AgeData(50));
19              heap.add(new AgeData(5));
20              heap.add(new AgeData(15));
21
22              System.out.println(heap.find(new AgeData(10)).toString());
23              System.out.println(heap.find(new AgeData(70)).toString());
```

Console

```
<terminated> Main (1) [Java Application] C:\Program Files\Java\jre1.8.0_251\bin\javaw.exe (3 May 2020 18:52:12)
10-2
70-1
```

There is some exception test below.

```
10      public static void main(String[] args) {
11          MaxHeap heap=new MaxHeap();
12
13          try {
14              heap.add(new AgeData(10));
15              heap.add(new AgeData(5));
16              heap.add(new AgeData(70));
17              heap.add(new AgeData(10));
18              heap.add(new AgeData(50));
19              heap.add(new AgeData(5));
20              heap.add(new AgeData(15));
21
22              //------------------EXCEPTION TEST ------------------
23              System.out.println("Does not exist age remove");
24              heap.remove(100);
25              System.out.println("Does not exist age find");
26              System.out.println(heap.find(new AgeData(150)).toString());
```

Console

```
<terminated> Main (1) [Java Application] C:\Program Files\Java\jre1.8.0_251\bin\javaw.exe (3 May 2020
Does not exist age remove
There is no this number age
Does not exist age find
There is no this number age in tree
```