fd

GIT Department of Computer Engineering

CSE 222/505 - Spring 2020

Homework #5 Part 4 Report

Murat YILDIZ

1801042004

PROBLEM SOLUTION APPROACH

STEP 1 – Implemented AgeData which holds age and number of people at that age, as Comparable

STEP 2 – AgeDataComparator which implements Comparator class , compare method takes two AgeData object and compare them according to number of people

STEP 3 – Implemented MaxHeap with Arraylist

STEP 4 – Implemented add(), remove(), find(), youngerThan(), olderThan() methods

STEP 5 – add() adds an AgeData at the end and move it according to number of people at that age
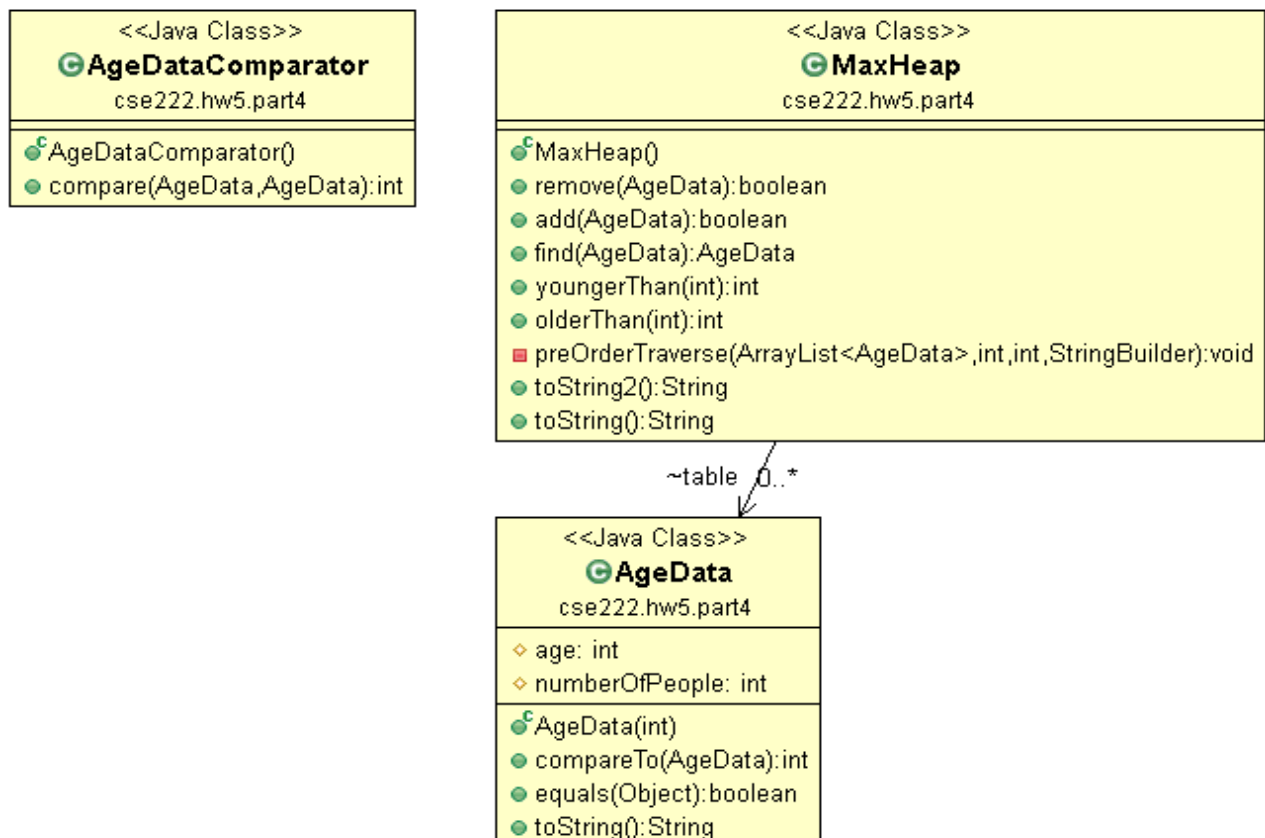
STEP 6 – remove() replace given AgeData and at the end of the heap then remove an at the end, then move it according to number of people at that age

STEP 7 – find() find data and return it

STEP 8 – youngerThan() find and print all data younger than given age

STEP 9 – olderThan() find and print all data older than given age

CLASS DIAGRAM

<<Java Class>>
ⒼAgeDataComparator
cse222.hw5.part4

- ⚫ᶜAgeDataComparator()
- ⚫ compare(AgeData,AgeData):int

<<Java Class>>
ⒼMaxHeap
cse222.hw5.part4

- ⚫ᶜMaxHeap()
- ⚫ remove(AgeData):boolean
- ⚫ add(AgeData):boolean
- ⚫ find(AgeData):AgeData
- ⚫ youngerThan(int):int
- ⚫ olderThan(int):int
- ◼ preOrderTraverse(ArrayList<AgeData>,int,int,StringBuilder):void
- ⚫ toString2():String
- ⚫ toString():String

~table 0..*

<<Java Class>>
ⒼAgeData
cse222.hw5.part4

- ◇ age: int
- ◇ numberOfPeople: int
- ⚫ᶜAgeData(int)
- ⚫ compareTo(AgeData):int
- ⚫ equals(Object):boolean
- ⚫ toString():String

TEST CASES

TEST ID	Scenario	Test Data	Expected Results	Actual Results	Pass/Fail
TEST01	MaxHeap()		Successfully Created	As Expected	Pass
TEST02	add(null)	null	Exception Throwed	As Expected	Pass
TEST03	remove(null)	null	Exception Throwed	As Expected	Pass
TEST04	find(null)	null	Exception Throwed	As Expected	Pass
TEST05	find(50)	50	Successfully Printed	As Expected	Pass
TEST06	remove(10)	10	Successfully Removed	As Expected	Pass
TEST07	remove(50)	50	Successfully Removed	As Expected	Pass
TEST08	remove(50)	50	Successfully Removed	As Expected	Pass
TEST09	Add(5)	5	Successfully Added	As Expected	Pass
TEST10	Add(5)	5	Successfully Added	As Expected	Pass
TEST11	youngerThan(50)	50	Successfully Returned	As Expected	Pass
TEST12	olderThan(11)	11	Successfully Returned	As Expected	Pass

RUNNING AND RESULTS

```
TEST01 - Creating a MaxHeap

Successfully Created
```

```
TEST02 - Testing add() with null

java.lang.NullPointerException


TEST03 - Testing remove() with null

java.lang.NullPointerException


TEST04 - Testing find() with null

java.lang.NullPointerException


After add some data respectively
(10, 5, 10, 70, 10, 50, 15, 15, 70, 50, 11)

10 - 2
15 - 2
70 - 2
50 - 2
5 - 1
11 - 1

10 - 2
15 - 2
70 - 2
50 - 2
5 - 1
11 - 1


TEST05 - Testing find() with 50

50 - 2
```

```
10 - 2
15 - 2
70 - 2
50 - 2
5 - 1
11 - 1
```

TEST06 - Testing remove() with 10

```
15 - 2
50 - 2
70 - 2
10 - 1
5 - 1
11 - 1
```

```
15 - 2
50 - 2
70 - 2
10 - 1
5 - 1
11 - 1
```

TEST07 - Testing remove() with 50

```
15 - 2
50 - 1
70 - 2
10 - 1
5 - 1
11 - 1
```

```
15 - 2
50 - 1
70 - 2
10 - 1
5 - 1
11 - 1
```

TEST08 - Testing remove() with 50

```
15 - 2
11 - 1
70 - 2
10 - 1
5 - 1
```

```
15 - 2
11 - 1
70 - 2
10 - 1
5 - 1
```

TEST09 - Testing add() with 5

```
15 - 2
5 - 2
70 - 2
10 - 1
11 - 1
```

```
15 - 2
5 - 2
70 - 2
10 - 1
11 - 1


TEST10 - Testing add() with 5

5 - 3
15 - 2
70 - 2
10 - 1
11 - 1



5 - 3
15 - 2
70 - 2
10 - 1
11 - 1


TEST11 - Testing youngerThan() with 50



5 - 3
15 - 2
70 - 2
10 - 1
11 - 1



TEST12- Testing olderThan() with 11

4
```