GTU Department of Computer Engineering

CSE 222/505 - Spring 2020

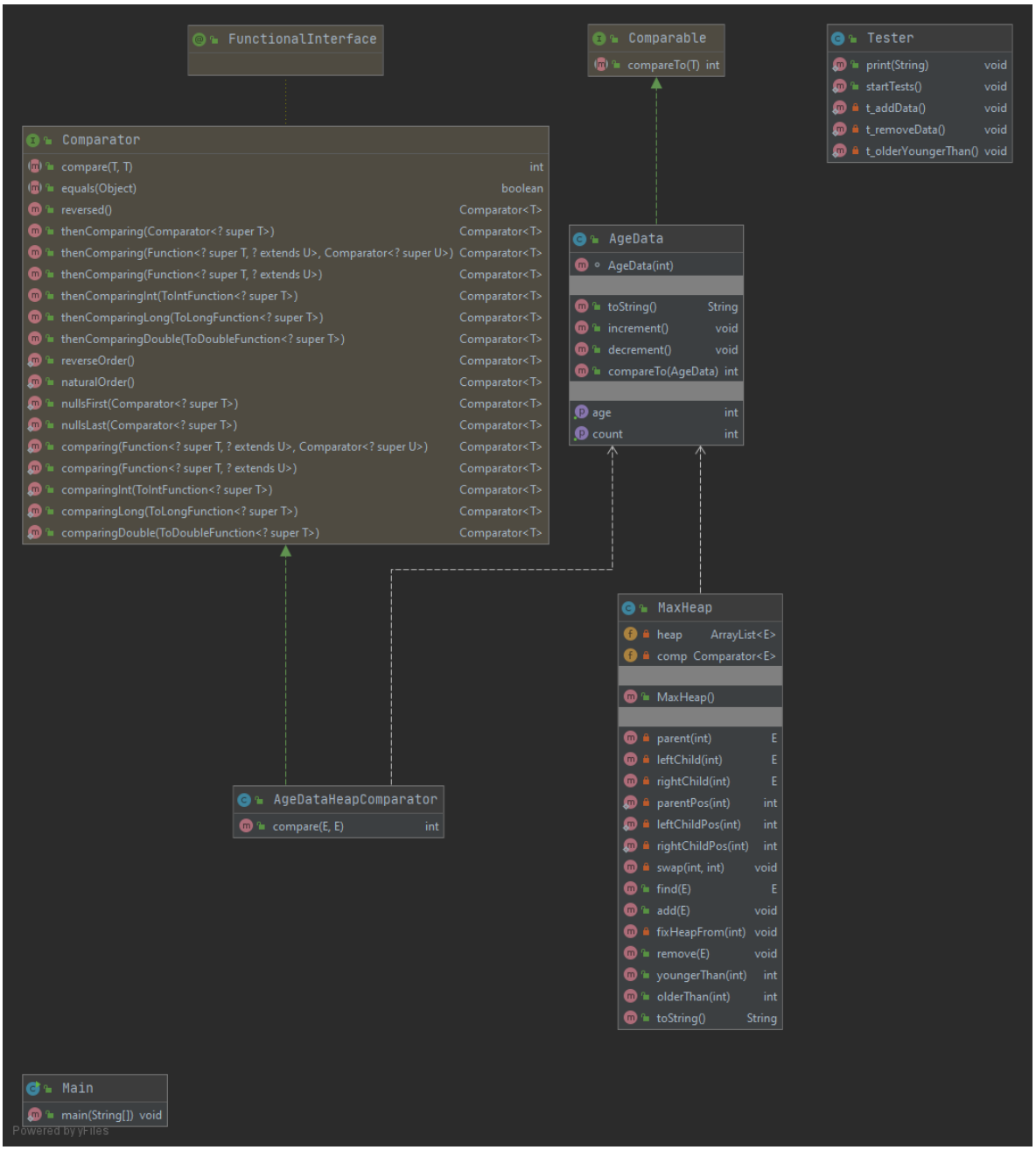
Homework 4 Report

Q4

Buğra Eren Yılmaz

1801042669

1. Class Diagrams



2.Problem Solution approach

The problem at hand was to create max heap structure, that Is primarily for age data, which consists of ages and number of people with that age.

First of I created the AgeData class which would be the main element of the max heap.

After that, I started creating the MaxHeap structure. First I started with abstractions of leftChild, rightChild, parents and those kind of things to find their position indexes in the array list that is used to represent the heap structure.

After that I created a fixHeap method that fixes the heap structure from given index and ensures that the heap structure never broke the max heap structure.

The adding and removing methods can broke the max heap structure so after these methods I always call the fixHeap method.

To make the max heap structures sorting factor to age count data, I created the age data heap comparator which implements java Comparator interface.

For youngerThan and olderThan methods, I traverse the heap arraylist fully, because heap is sorted in consideration of age counts, not the age it self.

3.Test Cases

Test ID	Test Scenario	Test Steps	Test Data	Expected Results	Actual Results	Pass/Fail
T01	Add age data as shown in given pdf	Call add methods for the given age datas	Empty age tree	heap with given datas	As expected	pass
T02	Removing age data to show that it decrements or deletes completely and also ensures that the max heap structure is	Construct initial tree, delete 10, delete 10, delete 20	Initial tree with data of, 10-2, 20-1, 5-1, 15-1	heap with data of, 5-1, 15-1	As expected	pass

	fixed after operation					
T03	Older than various tests	Construct initial heap, then calculate older than for, 5, 20, 10 and 0	Initial heap with data of, 10-2, 20-1, 5-1, 15-1	Correct older than counts for the given tree	As expected	pass
T04	Younger than various tests	Construct initial heap, then calculate younger than for, 5, 20, 10 and 500	Initial heap with data of, 10-2, 20-1, 5-1, 15-1	Correct younger than counts for the given tree	As expected	pass
T05	Find AgeData	Construct initial heap, then try to find AgeData with 15	Initial heap with data of, 10-2, 20-1, 5-1, 15-1	Will return the found 15 age data object	As expected	pass

4.Running results

Adding data to heap to demonstrate add and tostring functions:

10-2

5-2

70-1

50-1

15-1

Removing data from heap:

Initial heap:

10-2

5-2

70-1

50-1

15-1

Removing AgeData(10):

5-2

10-1

70-1

50-1

15-1

Removing AgeData(50):

5-2

10-1

70-1

15-1

Removing data from heap:

Initial heap:

10-2

5-2

70-1

50-1

15-1

olderThan(5): 5

olderThan(10): 3

olderThan(50): 1

youngerThan(5): 0

youngerThan(10): 2

youngerThan(50): 5

youngerThan(1000): 7

olderThan(0): 7

Finding data:

Initial heap:

10-2

5-2

70-1

50-1

15-1

find(AgeData(15)): 15-1