GTU Department of Computer Engineering

CSE 222/505 - Spring 2020

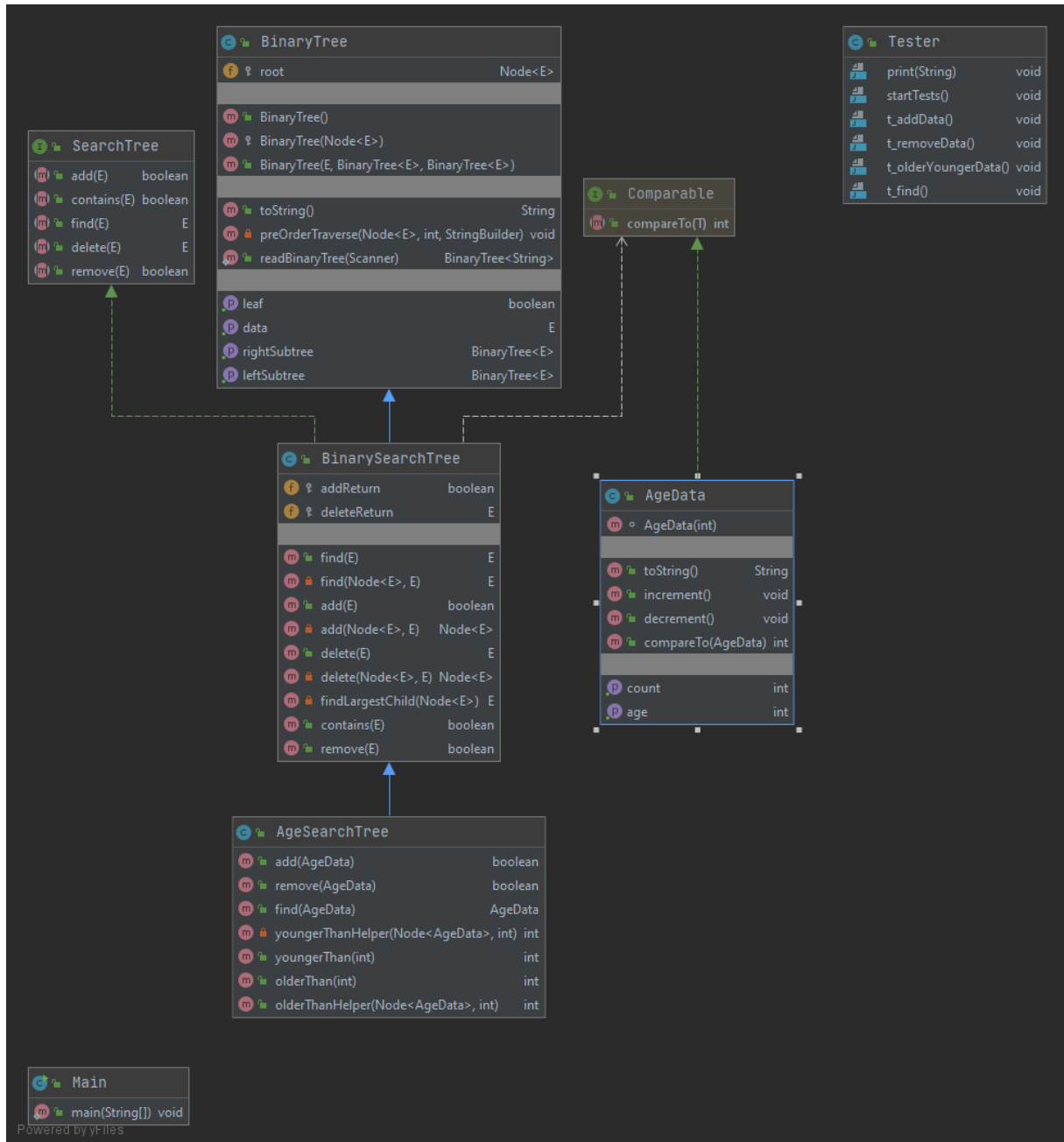# Homework 4 Report

Q3

Buğra Eren Yılmaz

1801042669

1. Class Diagrams

BinaryTree

root	Node\<E\>

BinaryTree()	
BinaryTree(Node\<E\>)	
BinaryTree(E, BinaryTree\<E\>, BinaryTree\<E\>)	

toString()	String
preOrderTraverse(Node\<E\>, int, StringBuilder)	void
readBinaryTree(Scanner)	BinaryTree\<String\>

leaf	boolean
data	E
rightSubtree	BinaryTree\<E\>
leftSubtree	BinaryTree\<E\>

SearchTree

add(E)	boolean
contains(E)	boolean
find(E)	E
delete(E)	E
remove(E)	boolean

Comparable

compareTo(T)	int

Tester

print(String)	void
startTests()	void
t_addData()	void
t_removeData()	void
t_olderYoungerData()	void
t_find()	void

BinarySearchTree

addReturn	boolean
deleteReturn	E

find(E)	E
find(Node\<E\>, E)	E
add(E)	boolean
add(Node\<E\>, E)	Node\<E\>
delete(E)	E
delete(Node\<E\>, E)	Node\<E\>
findLargestChild(Node\<E\>)	E
contains(E)	boolean
remove(E)	boolean

AgeData

AgeData(int)	

toString()	String
increment()	void
decrement()	void
compareTo(AgeData)	int

count	int
age	int

AgeSearchTree

add(AgeData)	boolean
remove(AgeData)	boolean
find(AgeData)	AgeData
youngerThanHelper(Node\<AgeData\>, int)	int
youngerThan(int)	int
olderThan(int)	int
olderThanHelper(Node\<AgeData\>, int)	int

Main

main(String[])	void

Powered by yFiles

2.Problem Solution approach

The problem at hand was to create a binary search tree that Is primarly for age data, which consists of ages and number of people with that age.

First of I created the AgeData class which would be a node of the future AgeSearchTree.

After that, I extended the BinarySearchTree implementation given in book and created the AgeSearchTree.

It is nearly identical to a general BST with only one change, the adding and removing may add or remove by incrementing or decrementing the age count of the nodes. To achieve this outcome I needed to override the add and remove methods.

For finding an AgeData, i just called the super.find because that does the same thing.

The traversals are all implemented for printing and debugging.

For youngerThan and olderThan methods, I traverse the tree with custom traversals. It does not traverse the whole tree, it terminates the traverse as soon as it met the conditions.

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	Age tree with given datas	As expected	pass
T02	Removing age data to show that it decrements or deletes completely	Construct initial tree, delete 10, delete 10, delete 20	Initial tree with data of, 10-2, 20-1, 5-1, 15-1	tree with data of, 5-1, 15-1	As expected	pass
T03	Older than various tests	Construct initial tree, then calculate older than for, 5, 20, 10 and 0	Initial tree 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	Construct initial tree,	Initial tree with data of,	Correct younger than	As expected	pass

	various tests	then calculate younger than for, 5, 20, 10 and 500	10-2, 20-1, 5-1, 15-1	counts for the given tree		
T05	Find AgeData	Construct initial tree, then try to find AgeData with 15	Initial tree 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 some data to demonstrate toString and add functions:

10-2

  5-1

    null

    null

  20-1

    15-1

      null

      null

    null

Removing AgeData to demonstrate decrementing:

Initial tree:

10-2

  5-1

    null

null

20-1

15-1

null

null

null

Removed AgeData(10):

10-1

5-1

null

null

20-1

15-1

null

null

null

Removed AgeData(10) again:

5-1

null

20-1

15-1

null

null

null

Removed AgeData(20):

5-1

null

15-1

null

null

Older/Younger AgeData counting:

Initial tree:

10-2

  5-1

    null

    null

  20-1

    15-1

      null

      null

    null

olderThan(5): 4

olderThan(20): 0

olderThan(10): 2

youngerThan(5): 0

youngerThan(20): 4

youngerThan(10): 1

olderThan(0): 5

youngerThan(500): 5

Finding AgeData:

Initial tree:

10-2

  5-1

    null

    null

  20-1

    15-1

      null

      null

Trying to find AgeData(15):

15-1