GIT Department of Computer Engineering

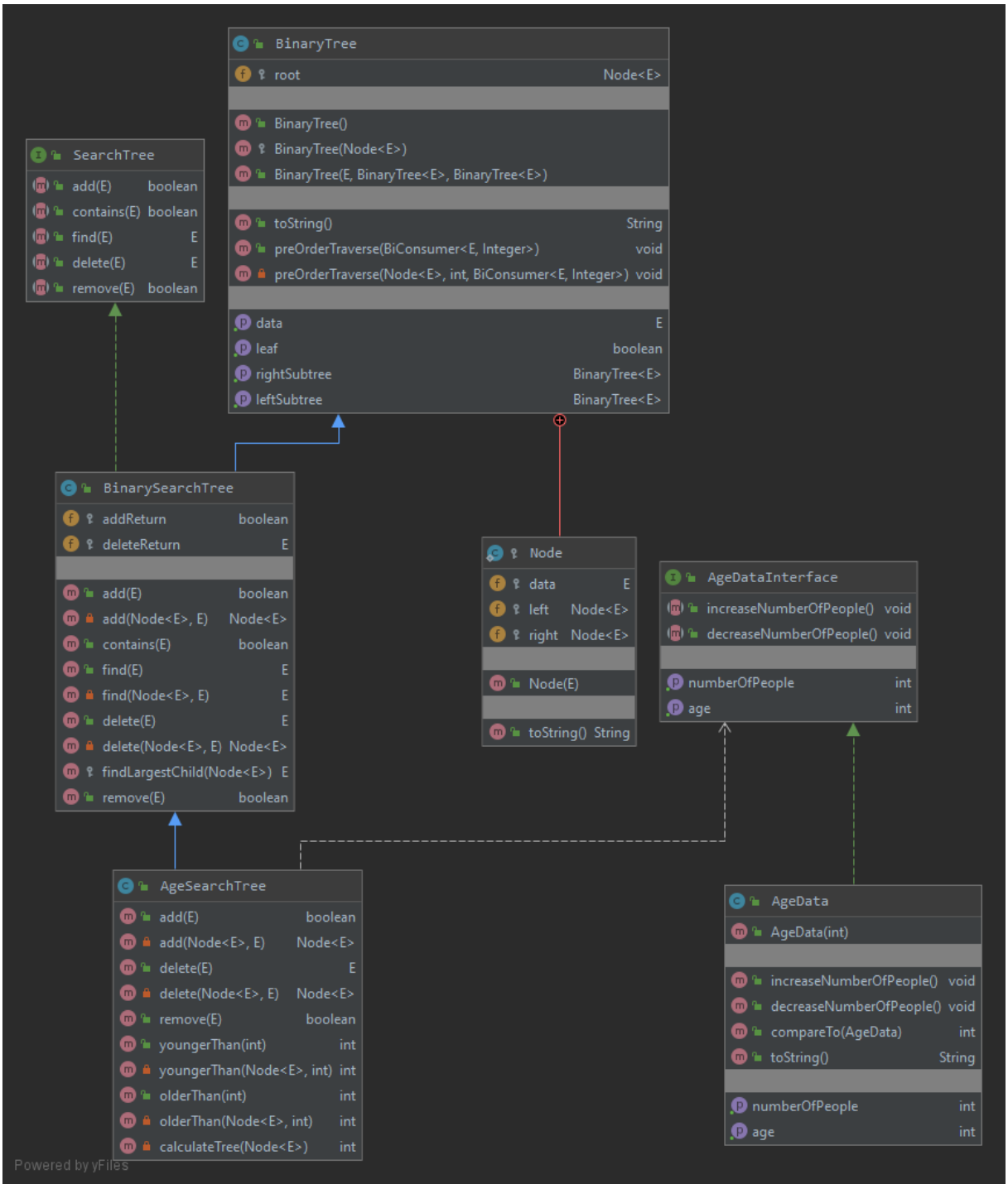
CSE 222/505 – Spring 2020

Homework #05 Part 3 Report

Abdullah ÇELİK

171044002

Class Diagram



Problem Solution Approach

Firstly, AgeSearchtree class is a generic class and generic type object should implement Comparable interface. I need to use this generic type method to be able to implement the methods of AgeSearchTree. I can do this in two ways. First, I can cast this generic object, but it doesn't disaccord object oriented programming. Second, I can force this generic type into an interface. In this way, while implementing the methods of the AgeSearchTree class, I can use the methods offered by this

interface. This path get along the object oriented programming. After doing this, implement the remaining methods with the correct algorithm.

Test Cases

Test ID	Scenerio	Test Data	Expected Results	Actual Results	Pass/Fail
TEST01	boolean add(E item) method called when tree is empty and has some elements	Tree Size : 0 Item : AgeData(10) Tree Size : 1 Item : AgeData(5) Item : AgeData(15) Item : AgeData(5) Item : AgeData(20) Item : null	Succesfully added except for null and returned correct boolean value	As expected	Pass
TEST02	boolean remove(E item) method called when tree has some elements	Tree Size : 4 Item : AgeData(15) Item : AgeData(10) Item : AgeData(5) Item : AgeData(5) Item : null	Succesfully removed except for null and returned correct boolean value	As expected	Pass
TEST03	E find(E e) method called when tree has some elements	Tree Size : 4 e : AgeData(20) e : AgeData(10) e : null	Succesfully returned correct value	As expected	Pass
TEST04	int youngerThan(int age) method called when tree has some elements	Tree Size : 4 age : 25 age : 10	Successfully returned correct value	As expected	Pass
TEST05	int olderThan(int age) method called when tree has some elements	Tree Size : 4 age : 5 age : 15	Successfully returned correct value	As expected	Pass

Running and Results

```
TEST01 - boolean add(E item)
When tree is empty, method will be called as
        tree.add(new AgeData(10))
Before adding
AgeSearchTree :
null

Add 10 : true
After adding
AgeSearchTree :
10 - 1
null
null

When tree has some elements, method will be called respectively as
        tree.add(new AgeData(5)), tree.add(new AgeData(15)),
        tree.add(new AgeData(5)), tree.add(new AgeData(20)),
        tree.add(new AgeData(null))
Before adding
AgeSearchTree :
10 - 1
null
null

Add 5 : true
Add 15 : true
Add 5 : true
Add 20 : true
Add null : false
After adding
AgeSearchTree :
10 - 1
5 - 2
null
null
15 - 1
null
20 - 1
null
null
```

```
TEST02 - boolean remove(E item)
When tree has some elements, method will be called respectively as
        tree.remove(new AgeData(15)), tree.remove(new AgeData(10)),
        tree.remove(new AgeData(5)), tree.remove(new AgeData(5)),
        tree.remove(new AgeData(null))
Before removing
AgeSearchTree :
10 - 1
5 - 2
null
null
15 - 1
null
20 - 1
null
null

Remove 15 : true
After removing
AgeSearchTree :
10 - 1
5 - 2
null
null
20 - 1
null
null
```

```
Remove 10 : true
After removing
AgeSearchTree :
5 - 2
null
20 - 1
null
null

Remove 5 : true
After removing
AgeSearchTree :
5 - 1
null
20 - 1
null
null

Remove 5 : true
After removing
AgeSearchTree :
20 - 1
null
null

Remove null : false
After removing
AgeSearchTree :
20 - 1
null
null

TEST03 - E find(E e)
When tree has some elements, method will be called respectively as
        tree.find(new AgeData(20)), tree.find(new AgeData(10)),
        tree.find(null)
AgeSearchTree :
20 - 2
5 - 1
null
15 - 1
null
null
25 - 1
null
null

Find 20 : 20 - 2
Find 10 : null
Find null : null
```

```
TEST04 - int youngerThan(int age)
When tree has some elements, method will be called respectively as
        tree.youngerThan(25), tree.youngerThan(10)
AgeSearchTree :
20 - 2
5 - 1
null
15 - 1
null
null
25 - 1
null
null

There are 4 people younger than 25
There are 1 people younger than 10

TEST05 - int olderThan(int age)
When tree has some elements, method will be called respectively as
        tree.olderThan(5), tree.olderThan(15)
AgeSearchTree :
20 - 2
5 - 1
null
15 - 1
null
null
25 - 1
null
null

There are 4 people younger than 5
There are 3 people younger than 15
```