Gebze Technical University
Computer Engineering

CSE 222 - 2018 Spring

HOMEWORK 03 REPORT

SİNAN ELVEREN
111044074

Course Assistant:
Fatma Nur Esirci, Tuğbagül Altan Akın, Mehmet Burak Koca

1 INTRODUCTION

1.1 Problem Definition

Part 1:

Construct a binary tree representation of general tree. Left root will childreen node and right root will siblings. Need to extend BinaryTree class for implement this tree. There are must have 2 methots for search metod. First one levelOrderSearch that travese tree as level by level and return found found node, second one postOrderSearch that traverse tree as post order way and return found node. Class also need to print to the tree as general tree representation(Override preOrderTraverse).

Part2:

Construct a general search tree structure where the tree includes multidimensional items. Each level of a multidimensional tree splits all children along a spesfic dimension.

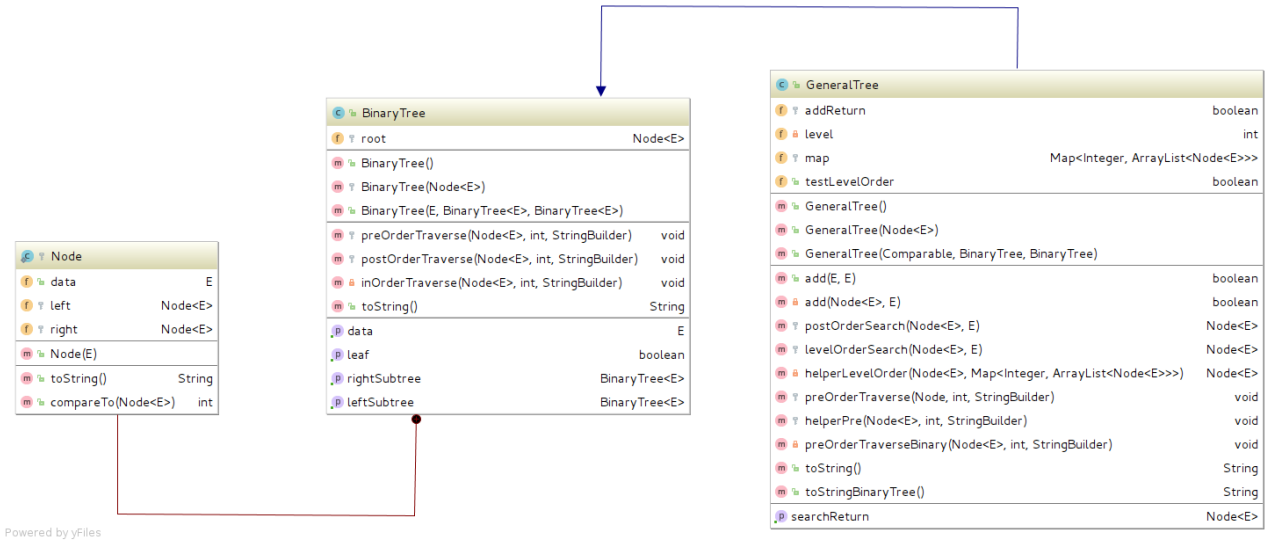
1.2 System Requirements

(for only part1) Program have 3 clases. First one is BinaryTree class . Second one is generalTree class that extended from BinaryTree class. In this class, you can add parents and childs to in tree,and you can show the tree binary or general tree representation. ToString() method is representing the tree as general tree representation. ToStringBinary() methoot is repreenting the tree as binary tree representation. LevelOrderSearch()methot is take 2 parameters that node and parent, so searching level by level and so returns found node. Similarly, postOrderSearch() method doing this but, it s searching as post order. Finally add method using this search methods and adding new node in tree.

I actually did not implement part 2, becouse of I have no time.

2 METHOD

2.1 Class Diagrams



3 classes for part 1. GeneralTree extends from BinaryTree and BinaryTree is including Node inner class.

2.2 Use Case Diagrams

There are no need Use Case Diagrams.

2.3 Other Diagrams (optional)

There are no diagrams more.

2.4 Problem Solution Approach

Firstly, I created a class named GeneralTree and extend it from BinaryTree. Add method is most important methot for design the general tree. Add method is have 2 paramter. First one is parent name, second one is child name. Add method using search methods as helper method for add new node in tree. Firstly, check child parameter and if it s uniq, call the search method via parent name for find the node and add it with child. Level order search is searching level by level and returning found node. Post order search is searching as post order and returning found node. If there are no parent, then returns null. Smilarly add method returns false if there are no addition on tree.

3 RESULT

3.1 Test Cases (for only part1)

Unit test:
Okay
Main test
Okay
BinaryTreeClass:
Already tested(this classfrom course's book)
GeneralTree Class:

toString() [General tree representation] tested, printing correctly

toStringBinary() [BinaryTree representation] tested, printing correctly

add() tested, addition have been done succesfuly

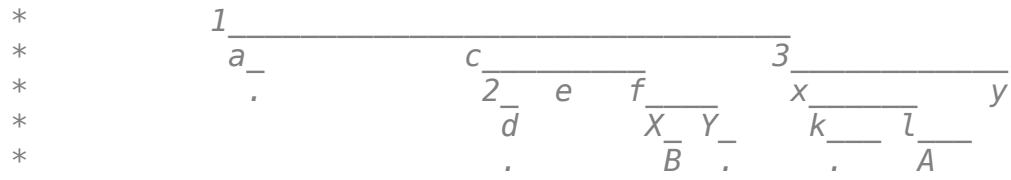
levelOrderSearch() already used in add and so tested, working correctly

postOrderSearch() already used in add and so tested, working correctly

3.2 Running Results

I tested step by step in main test.You can follow program's flow.

->Creaate GeneralTree, and test of part1
Add a tree node by node
General Tree representation(i.e “1” have thre child that “a ,c, 3”)



Follow it step by step.

```
    >General Tree<
1
   |a


Search in tree (Parent & Child) 2&b
(a)  (1)  (a)  (1)  nullfound

    >General Tree<
1
   |a


Search in tree (Parent & Child) 1&c
(a)  (1)  (a)  (1)  1found

    >General Tree<
1
   |a
   |c
```

```
testTree1.add("1", "a");

testTree1.add("2", "b");
//not inculde, NOP
//so null found

testTree1.add("1", "c");
...
```

```
Search in tree (Parent & Child) X&B
(a)  (d)  (2)  (e)  (X)  (Y)  (f)  (c)  (k)  (A)  (l)  (x)  (y)  (3)  (1)  (a)  (d)  (2)  (e)  (X)  Xfound
```

```
    >General Tree<
1
                                                    testTree1.add("X", "B");
   |a
   |c
   |    |2
   |    |    |d
   |    |e
   |    |f
   |    |    |X
   |    |    |    |B
   |    |    |Y
   |3
   |    |x
   |    |    |k
   |    |    |l
   |    |    |    |A
   |    |y
```

```
System.out.println(testTree1.toStringBinaryTree());
```

```
    >Binary Tree<
1
   |a
   |    |null
   |    |c
   |    |    |2
   |    |    |    |d
   |    |    |    |    |null
   |    |    |    |    |null
   |    |    |    |e
   |    |    |    |    |null
   |    |    |    |    |f
   |    |    |    |    |    |X
   |    |    |    |    |    |    |B
   |    |    |    |    |    |    |    |null
   |    |    |    |    |    |    |    |null
   |    |    |    |    |    |    |Y
   |    |    |    |    |    |    |    |null
   |    |    |    |    |    |    |    |null
   |    |    |    |    |    |null
   |    |    |3
   |    |    |    |x
   |    |    |    |    |k
   |    |    |    |    |    |null
   |    |    |    |    |    |l
   |    |    |    |    |    |    |A
   |    |    |    |    |    |    |    |null
   |    |    |    |    |    |    |    |null
   |    |    |    |    |    |    |null
   |    |    |    |    |y
   |    |    |    |    |    |null
   |    |    |    |    |    |null
   |    |    |    |null
   |null
```

```
System.out.println(testTree1.toString());
```

```
    >General Tree<
1
   |a
   |c
   |   |2
   |   |   |d
   |   |e
   |   |f
   |   |   |X
   |   |   |   |B
   |   |Y
   |3
   |   |x
   |   |   |k
   |   |   |l
   |   |   |   |A
   |   |y
```

Second Tree from Course's slide. General Tree Representation

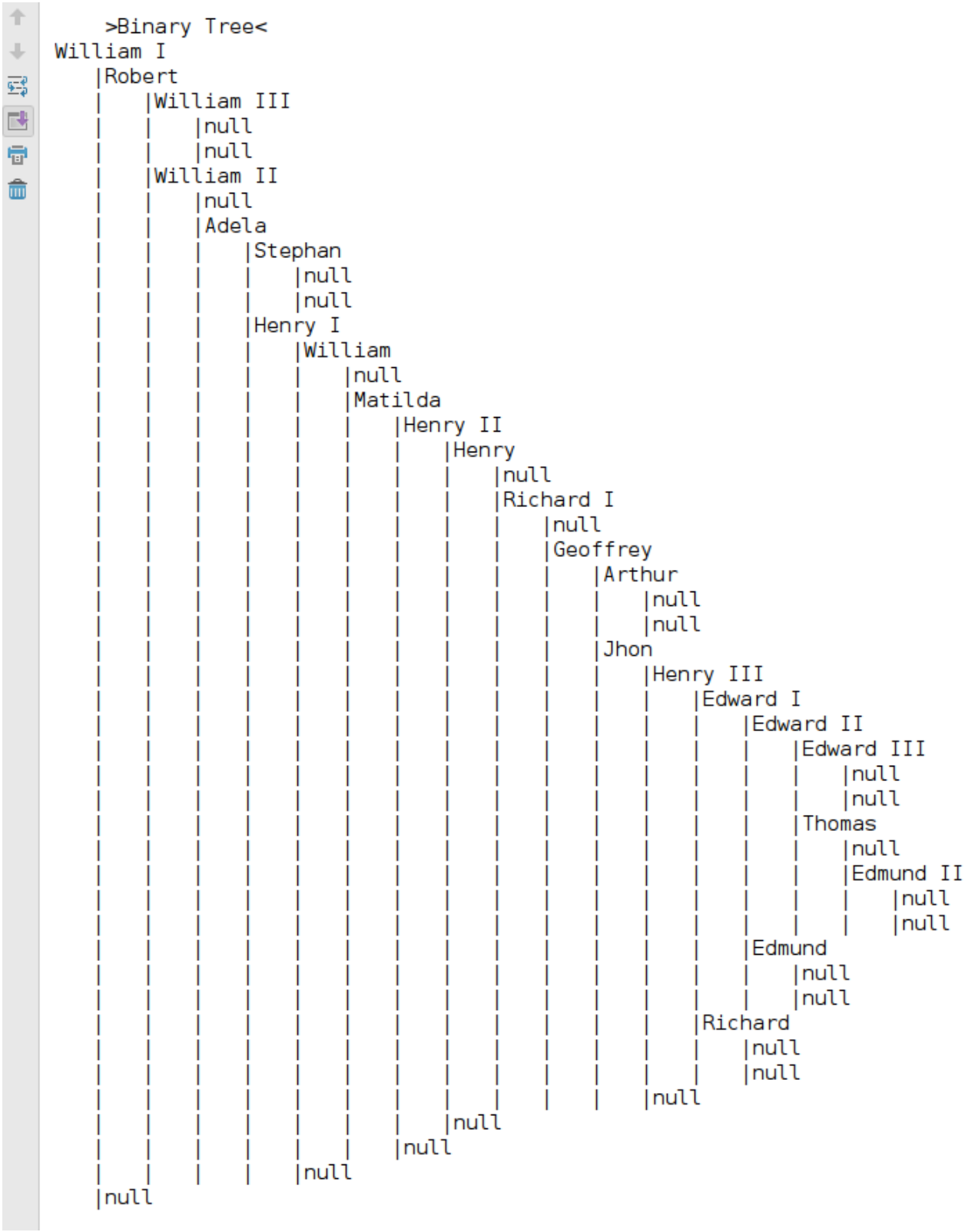
```
System.out.println(testTree2.toStringBinaryTree());
```

```
    >General Tree<
William I
   |Robert
   |   |William III
   |William II
   |Adela
   |   |Stephan
   |Henry I
   |   |William
   |   |Matilda
   |   |   |Henry II
   |   |   |   |Henry
   |   |   |   |Richard I
   |   |   |   |Geoffrey
   |   |   |   |   |Arthur
   |   |   |   |Jhon
   |   |   |   |   |Henry III
   |   |   |   |   |   |Edward I
   |   |   |   |   |   |   |Edward II
   |   |   |   |   |   |   |   |Edward III
   |   |   |   |   |   |   |Thomas
   |   |   |   |   |   |   |Edmund II
   |   |   |   |   |   |Edmund
   |   |   |   |   |Richard

Added Second Tree
```

Binary Tree representation.

```
System.out.println(testTree2.toString());
```



```
    >Binary Tree<
William I
   |Robert
   |   |William III
   |   |   |null
   |   |   |null
   |   |William II
   |   |   |null
   |   |   |Adela
   |   |   |   |Stephan
   |   |   |   |   |null
   |   |   |   |   |null
   |   |   |   |Henry I
   |   |   |   |   |William
   |   |   |   |   |   |null
   |   |   |   |   |   |Matilda
   |   |   |   |   |   |   |Henry II
   |   |   |   |   |   |   |   |Henry
   |   |   |   |   |   |   |   |   |null
   |   |   |   |   |   |   |   |   |Richard I
   |   |   |   |   |   |   |   |   |   |null
   |   |   |   |   |   |   |   |   |   |Geoffrey
   |   |   |   |   |   |   |   |   |   |   |Arthur
   |   |   |   |   |   |   |   |   |   |   |   |null
   |   |   |   |   |   |   |   |   |   |   |   |null
   |   |   |   |   |   |   |   |   |   |   |Jhon
   |   |   |   |   |   |   |   |   |   |   |   |Henry III
   |   |   |   |   |   |   |   |   |   |   |   |   |Edward I
   |   |   |   |   |   |   |   |   |   |   |   |   |   |Edward II
   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |Edward III
   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |null
   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |null
   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |Thomas
   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |null
   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |Edmund II
   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |null
   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |null
   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |Edmund
   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |null
   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |null
   |   |   |   |   |   |   |   |   |   |   |   |   |Richard
   |   |   |   |   |   |   |   |   |   |   |   |   |   |null
   |   |   |   |   |   |   |   |   |   |   |   |   |   |null
   |   |   |   |   |   |   |   |   |   |   |   |null
   |   |   |   |   |   |   |   |null
   |   |   |   |   |   |   |null
   |   |   |   |   |null
   |   |   |   |null
   |null
```

levelOrderTest(); follow search order.

```
testTree1.add("X", "B");
```

```
Search in tree (Parent & Child) X&B
[1]  [a]  [c]  [3]  [x]  [y]  [2]  [e]  [f]  [k]  [l]  [X]  Xfound

    >General Tree<
1
   |a
   |c
   |   |2
   |   |   |d
   |   |e
   |   |f
   |   |   |X
   |   |   |   |B
   |   |   |Y
   |3
   |   |x
   |   |   |k
   |   |   |l
   |   |   |   |A
   |   |y
```

- Main titles -> 16pt , 2 line break
- Subtitles -> 14pt, 1.5 line break
- Paragraph -> 12pt, 1.5 line break