

**Gebze Technical University
Computer Engineering**

CSE 222 - 2018 Spring

HOMEWORK 4 REPORT

**151044012
UĞURKAN ATEŞ**

M.Burak Koca

1 INTRODUCTION

1.1 Problem Definition

Part 1

In this part we are expected to implement a general tree. General tree's are kinda different from normal binary tree's, they have ability to hold multiple elements spawned from same root unlike binary tree(which only holds 2). This tree structure mostly used for familia structures. When identify a family tree big families like royal families etc have multiple branches and impossible to hold with binary tree. Problem expect us to implemen this kind of structure with binary tree structure.

Part 2

In this part we are given a MDS definition with examples. In multidimensional tree's tree structure shaped according to given parameters. This implementation also must come from binary tree structure also must have implement all search tree methods to work functionally. Such as find, findLargest etcetera.

1.2 System Requirements

Part 1

I needed a Binary Tree for later on i will extend this class on multiple circumstances. For this purpose Binary Tree class must work with Generic classes. It must only hold methods and variables will be used in all to most usages. So an add method is unnecessary to defined in this because all classes or parts have different add methods. This class also holds a inner Node class that will be super useful later on. I added methods and variables according to my needs in part 2 as well.

In part 1 i declared a class structure that is extended from binary tree and competable with generic structure. I have a preOrderTraverse method later on used in toString. This method recursively travels starting with root then left and right elements accordingly. While doing this addes their respective elements to stringBuilder object to print out later.

I needed adder method for add childs later on. And to child method work correctly i needed find method.

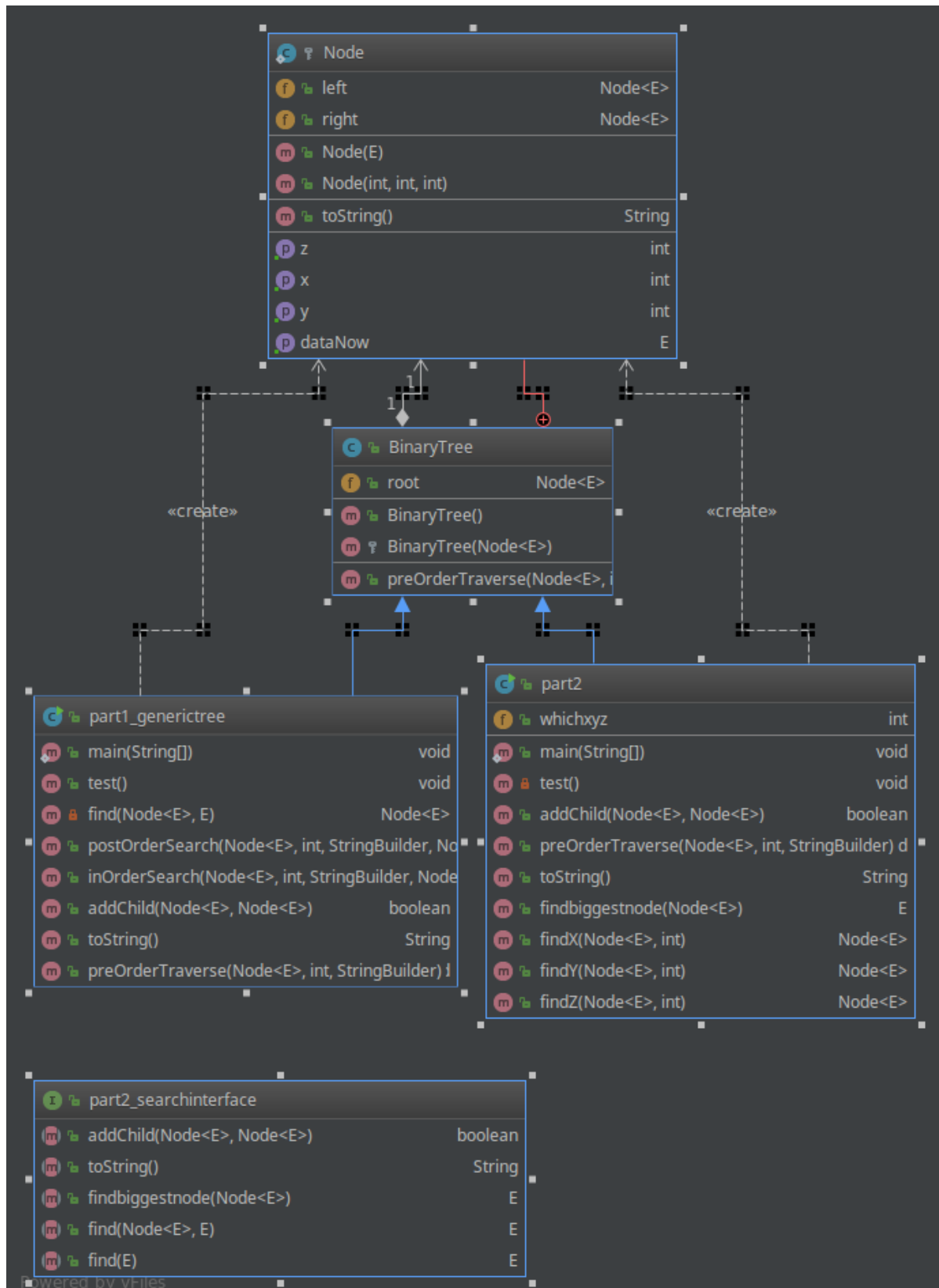
Part 2

I needed getter methods in Node structure for to add elements later on according to which element must be compared. I also added their values in class. WhichXYZ was for holding

which element must be compared at current time. Outside of these definition class structure is pretty similar to binaryTree class implementation which i extended. I also have to implement all SearchTree interface in this class.

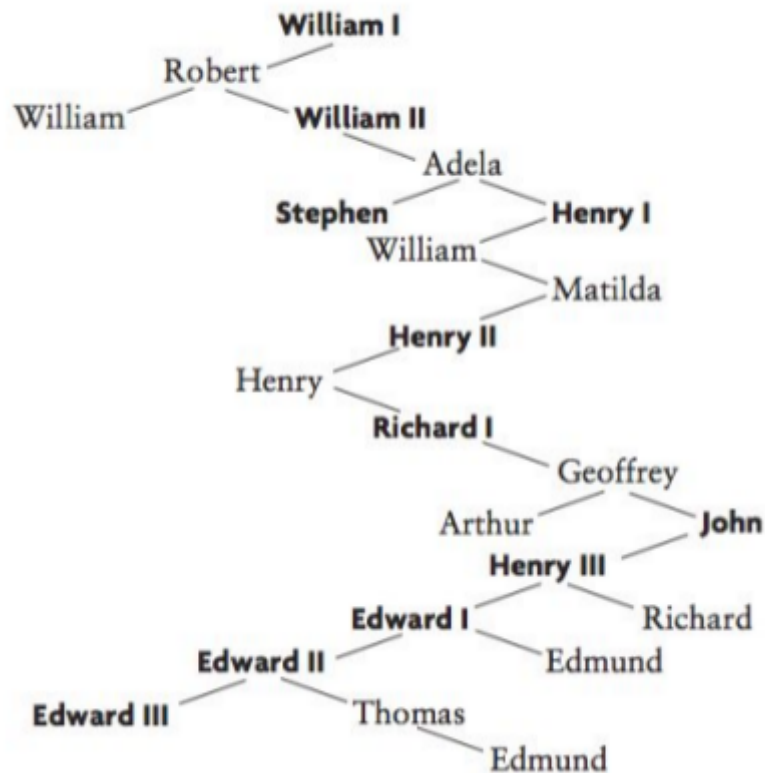
2 METHOD

2.1 Class Diagrams



2.2 Use Case Diagrams

I added test cases from the book as use case diagrams.(Implementation is pretty similiar)



(optional)

None Required.

2.3 Problem Solution Approach

For part 1 instead of making add method recursively i instead created a find method to find parent object i given with function call. Find method itself finds element recursively and return reference to it. When i find parent object (find operation is recursive and looks for elements recursively) then its pretty easy to add multiple items to structure. I look for left item if parent has any child if not i add to it. If it has element/s already then it must go right. Then it will traverse right childs when null appears. Then add to rightest element. ToString models use preorderTraverse.

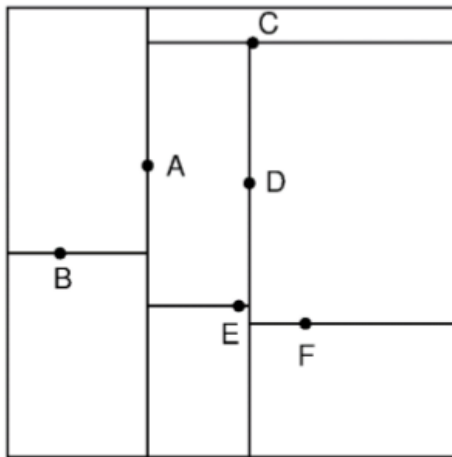
ForPart 2 i hold x,y,z and their respective getMethods to compare elements.I didnt use comperable<T> because i didnt find any use-case operation this might be useful other than numerical values.My tested approach is only useful for numerical values.

3 RESULT

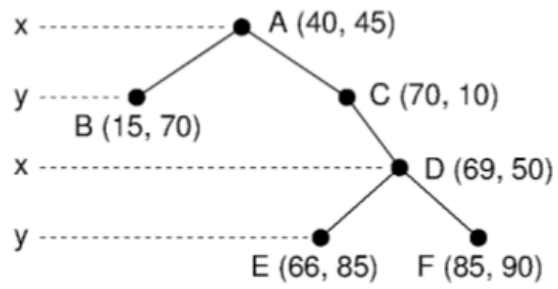
3.1 Test Cases

For part 1 i used Royal family of England to test my structure.

For part 2 i used PDF example of numbers. Both given in images.



(a)



(b)

3.2 Running Results

```
part1_generictree
/home/paypaytr/Documents/jdk1.8.0_131/bin/java ...
william1
  robert
    --
    --
  WILLIAM2
    --
  ADELA
    STEPHEN
      --
      --
  HENRY1
    william3
      --
      --
  MATILDA
    HENRY2
      HENRY3
        --
        --
      RICHARD1
        --
        GOFRET
          ARTHUR
            --
            --
          JOHN
            HENRY4
              --
              --
            --
      sonofRobert
        --
        --
Process finished with exit code 0
```