

**Gebze Teknik Üniversitesi**  
**Bilgisayar Mühendisliği**  
**CSE 222**  
**HW5**

**Hakkı Erdem Duman - 151044005**

## SORU 1:

İlk sorunun birinci kısmında kitaptan almış olduğum binary tree'nin içerisine bir add metodu, iteratör ve iki adet traverse metodu yazıp; oluşturulan ağacın pre-order şekilde traverse etmesini sağladım. Ağaç için bir doldurulma kuralı olmadığı için ağacı binary search tree kuralı ile doldurdum. Kitaptan almış olduğum BinaryTree class'ını da kendi kullanacağım şekilde değiştirdim.

Sorunun ikinci kısmında ise bir BinarySearchTree class'ı yazdım ve BinaryTree class'ını extend ettim. BinarySearchTree class'ının kendine özel bir iteratörü ve traverse metodu (level – order) var fakat ekleme metonodu BinaryTree class'ından alıyor çünkü BinaryTree class'ı da elemanlarını BinarySearchTree mantığı ile ekliyor.

## SORU 2:

İkinci soruda aile ağacını ilk çocuk sola, diğer çocuklar sağa gelecek şekilde oluşturudum. Eğer ekleme fonksiyonu, üç parametre de bir önceki node'lardan biri ile aynı olacak şekilde çağrılırsa veya olmayan bir parent ile çağrılırsa hata veriyor. Ayrıca kendim düzenleyebileceğim bir exception fırlatmak için MyException class'ını oluşturudum.

FamilyTree class'ı, traverse ederken BinaryTree'nin traverse metodlarını ve iteratörünü kullanıyor.

## Problem Çözümü

Birinci soruda, traverse methodunun içerisinde traverse ettiğim her elemanı, o traverse metodunun sırasına göre (pre-order veya level-order) bir ArrayList'in içerisine atıp o ArrayList üzerinde iterator kullandım. İlk soruda ArrayList kullanma sebebim buydu. Traverse metodunu da recursion yöntemi ile yaptım çünkü elemanları katlanarak artan bir yapıda döngü ile dolaşmak çok maliyetli olabiliyor.

İkinci soruda, ilk soruya ek olarak, valid ve invalid durumları kontrol etmek için Boolean array'i kullandım çünkü kullanılan olası bir boolean değişkeni, recursion biterken değişimini gerçekleştiremiyor. Array kullanıldığında ise array içerisinde değiştirilen bir eleman recursion'dan etkilenmediği için istenilen sonuç elde edilebiliyor.

FileOperations class'ını, main metodunda yapılan işi azaltmak için oluşturdum ve dosya okuma işlemini o class içerisinde yapıp, okunan değerleri bir ArrayList'e atıp, o ArrayList'i döndürüp, ArrayList'in içeriğini main metodunda değerlendirdim.

## KAYNAKÇA:

[https://www.tutorialspoint.com/java/java\\_string\\_trim.htm](https://www.tutorialspoint.com/java/java_string_trim.htm)

[https://en.wikipedia.org/wiki/Tree\\_traversal](https://en.wikipedia.org/wiki/Tree_traversal)

<http://stackoverflow.com/questions/20793082/java-comparing-generic-types>

<http://stackoverflow.com/questions/6271417/java-get-the-current-class-name>