

Gebze Technical University
Computer Engineering

CSE 222
2017 Spring

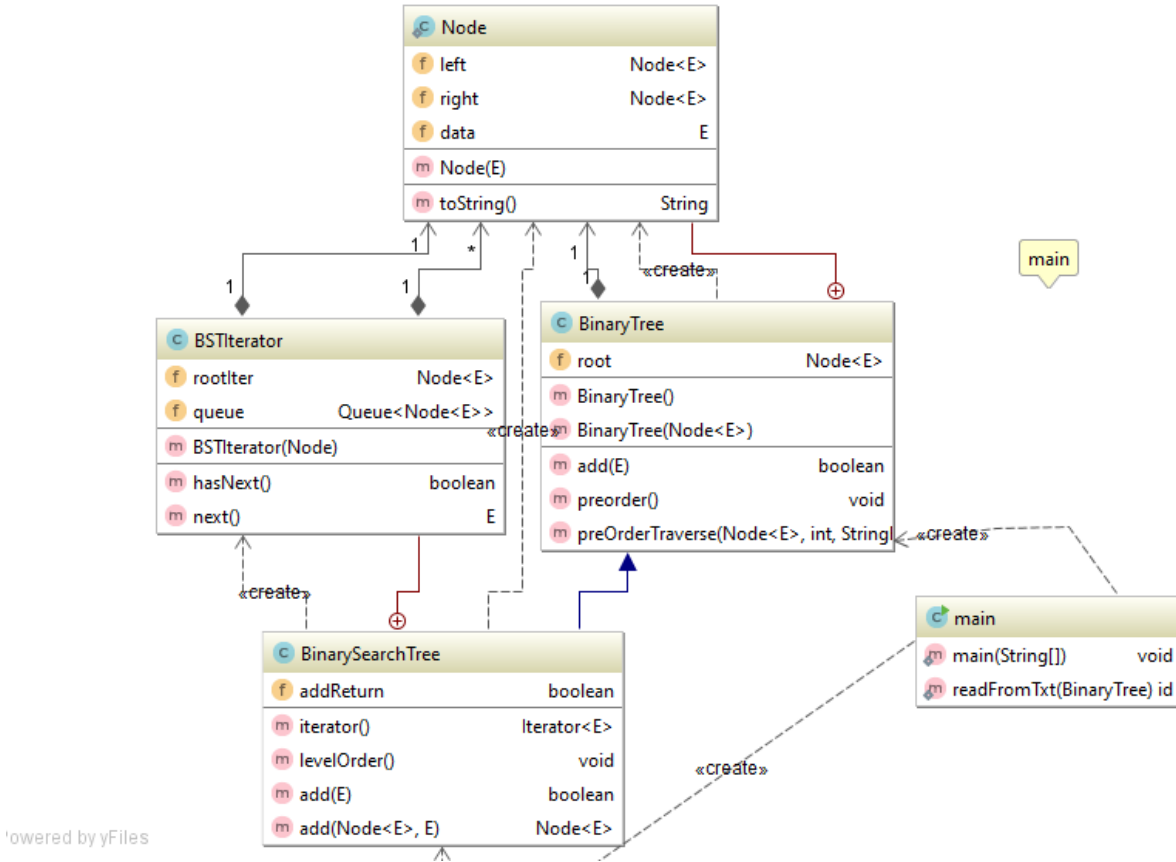
HOMEWORK 5 REPORT

Seyit Ahmet KARACA
141044084

Course Assistant:Nur Banu ALBAYRAK

Q1

1. Class Diagrams



2. Problem Solutions Approach

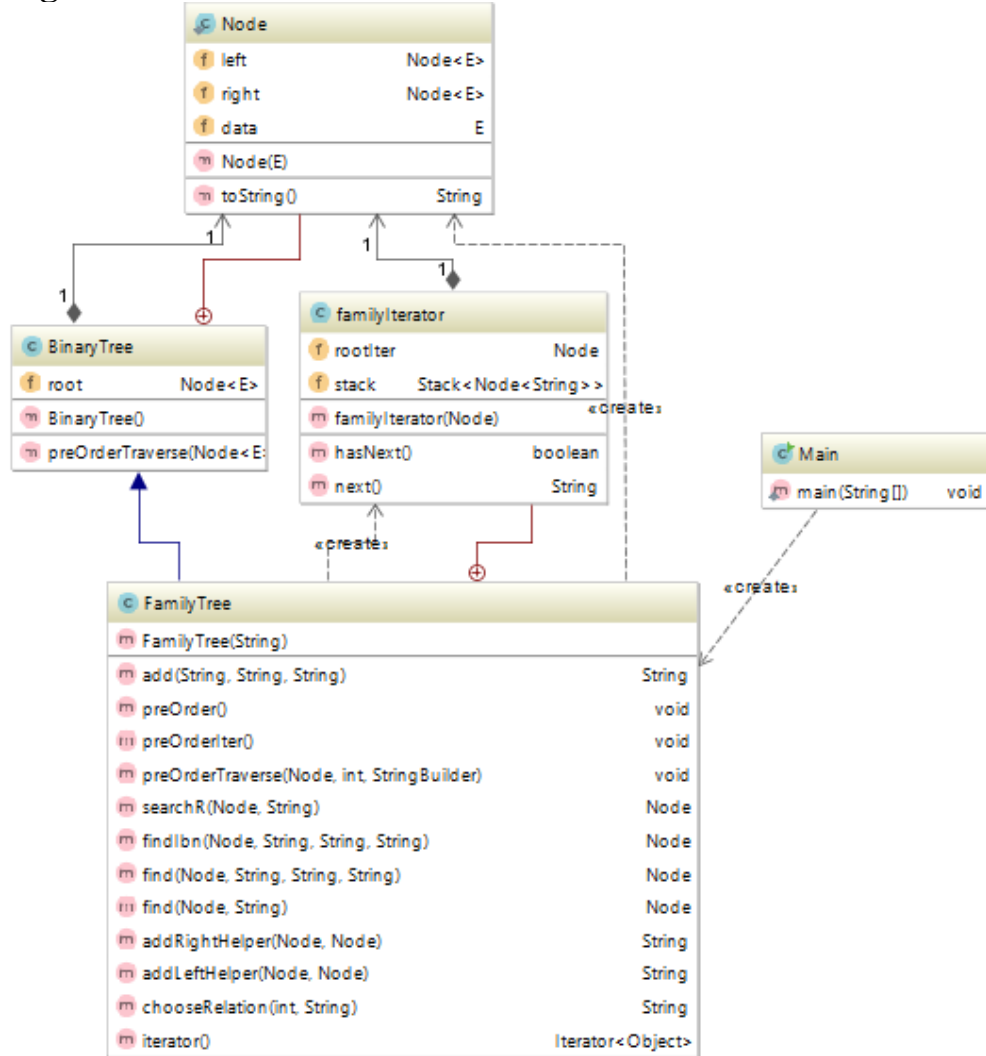
Problemde binary tree sınıfı oluşturmam gerekiyordu ve bu binary tree'de iterator ile ağacı pre-order gezdirmem gerekiyordu. Bunun için ilk başta binary tree için node ve binary tree ye eleman ekleyen kodları kitap yardımı ile implement ettim. Binary tree sınıfımda ekleme işlemi complete tree oluşacak şekilde ekleme işlemi yaptırdım. Bunun sebebi ise herhangi bir kural olmadığı için eklenmek istenen elemanları toplu olması daha mantıklı geldi. Implement ettiğim binary tree deki preOrderTraverse methodunu olduğu gibi kitaptakini kullandım çünkü ilk başta soru 2 den başladım ve değişikliği orda yaptım ve burda bir değişikliğe gerek kalmadı.

Not : 1.sorunun 1.partında iterator kullanmamışım. 1.sorunun 2.partında iteratoru implement etmişim.

Diğer sorun ise BinaryTree den binary search tree oluşturmamdı. Temel özellikler için bu sınıfı binaryTree sınıfımdan türettim. Üzerindeki tek değişiklik add methodu ve iterator eklemem oldu. Add methodu öncekinde complete tree oluşturacak şekildeyken burada elemanlar kök düğüme göre büyükler sağ tarafa küçükler sol tarafa eklendi. Iterator ile gezme ise inner iterator sınıfı içerisinde bir queue yardımı ile gerçekleştirdim. Iteratorun kurucu methodunda bu queue root ile ilklendiriliyor. Next methodu ilk çağrıldığında içerisindeki ilk eleman remove methodu ile alınıyor ve queue boşalıyor. Çıkarılan elemanın sol ağacı boş değil ise sıraya ekleniyor sonrasında ise sağ ağaç boş değil ise sağ ağacı sıraya ekleniyor. Tekrar next methodu çağrıldığında sıradaki eleman için aynı işlemler gerçekleştiriliyor ve level order olarak ağaç gezilmiş oluyor..

Q2

1. Class Diagrams



2. Problem Solutions Approach

İlk önce kitaptaki family tree yi anlamam gerekti. Root eleman aile reisi sol ağacında çocuğu sağ ağacında ise kardeşi bulunacak şekilde ağaca ekleme yapılmıştı. Sonraki adımda binary tree yi kitaptan aldım. FamilyTree'yi binaryTree'den türettim ve iterable interface'ini implement ettim. Sonraki asıl sorun "ibn-Fatih" ve "ebu-Fatih" gibi bana hiç yakın olmayan kelimeleri anlamaktı. Add methodu 3 parametre alıyor ve ilk parametre eklenecek insanın ismi , ikincisi ebeveyninin ismi ve üçüncüsü ise değişebiliyordu. Üçüncü parametre "ibn" ile başıyorsa ilk önce ağaçta ibn den sonra gelen ismi aradım onun sol ağacındaki node ve onun sağındakilerde ebeveyni aradım eğer varsa bulduğum node'un soluna çocuk olarak ekledim. Yani yaptığım işlem önce 2 nesil üste baktım. Bulduğunda çocuğuna ve çocuğunun kardeşlerine baktım var ise bulduğunun çocuğu diyerek ekledim. Bulamadı ise diğer ağaçta ibn ile başlayan ismi aradım ve hala yoksa hata verdirdim, bulursa alttaki node'larda eklettirdim. Diğer sorun ise "ebu" ile nickname i çözmekti. Bunda ise söylenmek istenen şey ebu dan sonra gelen isim kimin ebeyvni olduğu yani bu sorunu çözmek için kullandığım yol ise ; ilk önce parentname i aradım ağaçta ve bulunca ebu'dan sonra gelen isme baktım. Bu ebu'dan sonra gelen isim çocuklarından biri ise o ebeveyne birinci parametre

ile gelen ismi parentname olarak gelen node'a çocuk olarak ekledim. Yani ebu'dan sonra gelen isim ilk parametre ile aynı ise ebeveyne direkt olarak ilk parametre ile gelen insanı ekledim, ebu'dan sonra gelen isimle ilk parametre farklı ise ebeveyni bulunca ebu'dan sonra ne yazıyorsa ona ilk parametre kardeş olarak ekledim.

Traverse methodunu ktiaptan esinlererek implement ettim. Methoddaki değişiklik ,ebeveynler aynı seviyede çocuklar bir boşluk içeride olacak şekilde ekrana yazdırıldı.

Iterator ile aileyi aile reisi , çocuğu ve varsa çocukları olarak yazdırdım. Bunu girintili bir şekilde yazdıramadım çünkü iteratorde sadece Next methodunu kullandığım için herhangi başka bir parametre veremedim o yüzden alt alta yazdırmak zorunda kaldım. Bu implementasyonu yapmak için iteratorun içerisinde stack kullandım. Stack i ilklendirmek için ağacın rootunu kullandım. Next methodu ilk çalıştığında rootu stackten alıyor ve rootun önce sağ sonra sol ağacını stacke ekliyor. Stack ten pop yapınca en üstteki elemanı verdiği için sürekli sol eleman en üstte olacak ve sol ağaçlar null olduğunda sağ ağaçlar stacke eklenmeye başlanacak yani buda ebeveyn varsa çocuğu varsa onların çocukları yoksa kardeşleri ve onların çocukları olarak ekrana basacak.