

MUZAFFER METEHAN ALAN – 151044038

CSE222 - SPRING 2017

HW5

Genel Bilgiler :

- Öncelikle verilen soruların gerekli analizlerini yaptım , kitaptan Tree konusunu okudum.
- Daha sonra kitaptaki Binary Tree kodunu aldım, kodda eksik olan belirli fonksiyonlar vardı.
- Bunlar : Insert , Depth , Iterator classını implement edecek olan class.
- Daha sonra ödevde istenilen Pre order Traverse methodunu , iterator yardımıyla implement ettim.
- 1.sorunun 2.şikkı olaran BinarySearchTree için BinaryTree classını extend ettim.Iteratorunu ve Traverse methodunu override ettim.Traverse methodu level order şekilde isteniyordu.
- Daha sonra bu iki class'a ait fonksiyonların test edildiği main classını yazdım.Basit bir test yaptım ve sonucunu screenshot olarak ekledim.
- 2.Soru için kitapta okuyup anladığım FamilyTree algoritmasını uyarladım.
- BinaryTree classını extend ettim fakat ek olarak Node classına bir data daha ekledim (nickname) tutulması için.
- Daha sonra kendim bir MyException classı yazdım , exception fırlatılması gereken durumda bu classı kullandım.
- Family'e eklemek için kullanılan insert methodunu implement ettim.

- Mainde Pdfte verilen testi denedim ve traverse fonksiyonuyla ekrana bastırdım , sonucu screenshot olarak ekledim
- Aynı şekilde invalid bir durum olan , aynı parenta aynı isimde bir çocuk daha ekleme durumunu da test ettim, exception fırlatılma mesajıyla birlikte screenshot ekledim.
- JavaDoc dökümanları hepsi için yapıldı ve eklendi ,class Diagramları Enterprise Architech'te yapıp eklendi.

Problem Solution Approach:

İlk olarak kitaptan aldığım Binary Tree kodunu inceledim analizini yaptım nasıl çalıştığına dair mantık yürüttüm.Daha sonra travers işleminin nasıl yapılacağını dair algoritmalar tasarladım

Recursive veya iterative olarak da yapılabileceğine karar verdikten sonra kitapta recursive olarak bulunduğu için ben iterative olarak yazmaya karar verdim.Stack ve ArrayList gibi yapılardan da yardım alarak iterative olarak PreOrderTraversi yazdım.

Daha sonra kodda insert fonksiyonu olmadığı için kendim bir insert kodu yazdım.Insert kodu , sola kendisinden küçük value olanları ve sağa kendisinden büyük value olanları koyacak şekilde çalışıyordu.

BinaryTreeSearch classı için LevelOrderTravers algoritmasının çalışma mekanizmasını anladım (internette bulduğum pseudo code algoritmasını dile implement ettim.)

Insert fonksiyonum zaten BinarySearchTree fonksiyonu için de valid olarak çalıştığı için tekrar insert fonksiyonu eklemedim.

Bu 2 Travers methodunu test edecek olan , test.txt dosyasından integer okuyarak ağaç oluşturacak olan main classını yazdım.

Main classında bu 2 methodu çağırdım ve test ettim.2.soru için kitapta bulunan , family tree algoritmasını kafamda tasarladım ve bu algoritmaya uygun insert methodunu implement ettim.Eğer eklenecek olan kişinin parent ı ve nick 'i aynıyken , aynı kişi eklenmeye çalışılırsa exception fırlatılmasını gerçekleştirdim.Onun dışında yeni olarak eklenen her child , eğer ilk child ekleniyorsa parentın sağına daha sonra eklenecek olan childlar ilk eklenen childın soluna eklendi.

Daha sonra FamilyTree classını test edecek olan main classını implement ettim , pdfte verilen testi uyguladım.