

B141210019-B141210087 RAPOR

ÖDEVDE BİZDEN İSTENEN

İkili arama ağacını , Dugum sınıfını ve Bağlı liste sınıfını kullanarak ve verileri dosyadan çekerek bir ağaç tasarlamamızı istenmiş.Aynı zamanda bu ödevde kendimize ait sınıfların kütüphanesini de oluşturarak MinGW derleyicisinde derlememiz istenmiş.

ÖĞRENDİKLERİMİZ

Bu ödevde ikili arama ağacının yapısını öğrendik . Nasıl ağaca eleman ekleneceğini ve bunun nasıl gerçekleşeceğini öğrendik.Aynı zamanda ağacın dugumunun içinde Bağlı liste tutmayı öğrendik. Tüm bunları dosyalamayla nasıl birleştireceğimizi de öğrendik.İkili arama ağacının çok hızlı arama yapma özelliğini görmüş olduk.

ÖDEVDE YAPTIKLARIMIZ

Ödevin başlangıcında her şey den önce Node sınıfımızı oluşturduk ve Bunun Kurucusunu Binary Search Tree için uygun hale getirdik ve ona uygun şekilde Node sınıfının objelerini oluşturduk.

Daha sonra Binary Search Tree sınıfımızı oluşturduk. Bu sınıfta sadece bu ödev için gerekli olan Ekleme ve PreOrder İnOrder PostOrder metotlarımızı oluşturduk. Bunun yanı sıra bu metotların herkese erişilebilir kısmında mainden daha basit bir arayüzle bu işlemleri gerçekleyebilelim diye oluşturduk.Ekleme sınıfını oluştururken dikkat etmemiz gereken şeyler vardı. Bu ödevde verilerin önce ağaçadı daha sonra dugum adı çekilecekti. Yani ekleme işlemini bu halde yapacak olursak else komutuna girecek ve hiçbir ekleme işlemi yapmayacaktı. Bu yüzden ekleme metodumuzu bu hale uygun getirmek için aynı ağaç isminde olanların elemanlarını bir bağlı listeye aktarmamız gerekiyordu. Liste sınıfımızı ve metotlarımızı oluşturduk.Bu listeyi parametre olarak Node sınıfımıza eklemiş olduk. Böylece ekleme işleminin yapacağı iş Asıl Ağaç adı olarak ilk veriyi çekecek ve eğer aynı ağaç adından birden fazla rastgeldiği her seferinde o ağaca ait olan listeye gidecek be oraya dugumun içeriklerini yazmış olacaktır.Böylece ekleme işleminide tamamlamış olduk.Preorder İnorder Postorder işlemleri daha kompleks bir yapıda olmalıydı.Çünkü artık veri olarak çekeceği şey ağacın adı değil aynı zamanda o dugumun içindeki mevcut listenin içerisindeki dugum verilenide çekmeliydi. Onun için Liste sınıfında display adında yeni bir metot oluşturduk.Bu yeni metot listenin içindekileri head i kaybetmeden bir döngü içerisinde tüm liste elemanlarını ekrana döndürüyordu.Daha sonra ekran görüntüsünü aldıktan sonra MinGW derleyicisinde projemizi derledik ve exe. dosyasını elde etmiş olduk.

EKSİK BIRAKTIĞIMIZ YERLER

Ödevde galiba bizden Liste içerisindeki elemanlarında preorder inorder ve postorder olarak sıralanmasını istiyordu.Bu işlemi yapmak için çoğu fonksiyonun override halini oluşturmamız gerekiyordu.O işlemleri tamamen hallettikten sonra listede bir metot oluşturduk ve ana mantık listenin içindeki verilerle içinde bağlı liste bulunmayan yeni bir ağaç oluşturmaktı. Ve kompleks olmayan override İnOrder PreOrder ve PostOrder metodunda basit bir şekilde sıralayacaktık. Ama nedense Liste sınıfında include"BST.hpp"

olarak belirtmemize rağmen BST eklediğimiz an BST sınıfında 200 hata aldık. Mantığını doğru düşündüğümüzü düşünüyoruz ama bunu koda yansıtamadık.

ZORLANDIĞIMIZ YERLER

Yukarıdada açıklamasını yaptığımız gibi bizim zorlandığımız yer Liste içerisindeki elemanlarıda InOrder PreOrder ve PostOrder şekilde sıralamaktı.