

02.09.2020

Veri Yapıları Ödev-2

Kullanıcı Bilgilerini İkili Arama Ağaçları
ve Stack Yapıları İle İşleyen Program

İsim: Osman Kurt

No: G161210374

Şube: 1-A

İkili Arama Ağacı ve Stack yapılarını kullanarak, verilen dosya üstünden, kullanıcı bilgilerini # karakterine göre ayrıştıran ve ayrıştırılan bilgileri, User nesnesi olarak tanımlayan, sonrasında bu bilgileri işleyen program. Bu programı geliştirirken birden fazla veri yapıları ders konseptini gerektiren yapılar mevcuttu. Örnek vermek gerekirse bağlı listeler, yığıt yapıları, ikili arama ağaçları vs.. Bu yapıları gerçeklemek için 4 adet yardımcı sınıf kullandım. Odaklandığım yapılardan biri Stack yapısıydı ve bu yapı için, StackNode ve Stack adında iki sınıf tanımladım. StackNode, Stack yapısında bulunacak düğüm türünü temsil ederken, Stack ise Stack yapısı ile alakalı gerekli işlemleri yapan sınıftı. Bunlarında dışında bir diğer yapı ise Binary Search Tree idi, ben yazmış olduğum kodda bu yapıyı BST ve BSTNode sınıfları ile gerçekledim. Yine BST, ikili arama ağaçları ile ilgili gerekli işlemleri yapan kısım, BSTNode ise ikili arama ağacındaki düğümleri temsil eden sınıftı.

Uygulama ilk açıldığında, “Kisiler.txt” dosyası, tanımlamış olduğum, dosya okuma işlemlerini yapan FileManager sınıfı içinde ki “readAndProcessFile” fonksiyonu ile gerekli dosya okuma işlemleri başlıyor. Sonrasında dosyadan okunan her bir satır, “processUserData” adında ki BST sınıfında bulunan bir fonksiyonu, kullanıcı verilerini ayrıştırmak için yollanıyor. Gerekli ayrıştırma işlemleri bitince, user nesnesi oluşturuluyor ve bu veriler user nesnesine atılıyor. Sonrasında toplanan bu gerekli veriler ile, ikili arama ağacı yapısına uygun olarak “insertNodeToBST” fonksiyonu içerisinde atama işlemleri yapılıyor.

```
C:\Users\Mr.robot\Desktop\Veri Yapıları - 2>make
g++ -I ./include/ -o ./lib/BST.o -c ./src/BST.cpp
g++ -I ./include/ -o ./lib/BSTNode.o -c ./src/BSTNode.cpp
g++ -I ./include/ -o ./lib/FileManager.o -c ./src/FileManager.cpp
g++ -I ./include/ -o ./lib/Stack.o -c ./src/Stack.cpp
g++ -I ./include/ -o ./lib/StackNode.o -c ./src/StackNode.cpp
g++ -I ./include/ -o ./lib/User.o -c ./src/User.cpp
g++ -I ./include/ -o ./bin/Test ./lib/BST.o ./lib/BSTNode.o ./lib/FileManager.o ./lib/Stack.o ./lib/StackNode.o ./lib/User.o ./src/Main.cpp
./bin/test
Leyla,1992,60 Yigit...:
Mehtap,1987,74 Yigit...:
Murtaza,1982,85 Yigit...:
Ahmet,1980,72 Yigit...:
Mehmet,1974,82 Yigit...:
Press any key to continue . . .
```

Programın derlenmesi üzerine ekran görüntüsü

Not: AVL ağacı konseptini ödevime uyarlayamadığım için, ödevimi dengeleme işlemlerinin yapılmadığı versiyonu ile yolluyorum. Ama ikili arama ağacını oluşturan düğümlerde, bizden oluşturmamız istenen, kullanıcı ve Stack yapısı mevcut. Stack yapıları ve ikili arama ağacı doğru çalışıyor, uygulama sonlandırılmadan öncede ağaç üstünde Inorder travel yapılıyor ve sonuç ekrana bastırılıyor.