

Öğrendiklerim: Bağlı Listelerin yapısını kavramama yardımcı olduğunu söyleyebilirim. Özellikle büyük programlarda ya da yapmış olduğumuz ödev gibi komplike uygulamaları yazarken Bağlı Liste yapısı oldukça kafa karıştırıcı ve beyin yorucu olabiliyor. Ödev hem kafa karıştırırken hem de gayet açık ne yapılabileceğini gösteriyor. Gerisi ödevi yapan öğrenciye kalıyor. Pointerların kullanımı, referansların kullanımı, classların kalıtım yoluyla aktarılması, uygulama çalışma algoritması vs. çok şey pekiştirdim. Ödevin bana fazlasıyla yararı dokundu diyebilirim.

Ödevde Yaptıklarım: Ödevi kod üzerinde yazmadan önce kağıt üzerinde taslağını çizdim. Kafamda bir yol haritası oluşturduktan sonra koda geçtim. İlk olarak dosya hiyerarşilerini, header ve cpp dosyalarını oluşturdum. Kağıt üzerindeki taslağımda düşündüğüm dosyalar benim ihtiyacım olan kadardı. Proje sonuna kadar başka bir dosya eklemesi yapmadım. Koda bağlı listeleri oluşturarak başladım. İlk önce Dugum ve SatirListesi'nden başlasamda aslında YoneticiListesi'nden başlayarak ve Dugum'e inerek tanımlamaları yapmam gerektiğini anladım. Listeler arasında şimdilik kurabildiğim temel bağlantıları oluşturdum. YDugum ile SatirListesi bağlantısını oluşturmak için SatirListesi nesnesi tanımlamalıydım. Ancak daha SatirListesine verileri aktarmış değildim. Bu yüzden DosyaOkuma class ını oluşturmaya başladım. Kurucusu ile birlikte satirOkuma ve satirSayisiBulma fonksiyonları ile dosya okuma işlemlerinin temelini atmış bulundum. SatirListesi'ne satırları aktarabilmek için satirOkuma'da okunan satırları bir string dizisine atadım. Yine DosyaOkuma içerisinde float türünde, stringlerin ortalamalarını hesaplayıp satırlarla aynı index orantılı tutan bir dizi oluşturdum. Artık satırlar ve ortalamalarını depolamıştım. SatirListesi'ne aktarmadan önce ortalamalarına göre küçükten büyüğe sıralama yapmamı sağlayacak bir void fonksiyon tanımladım. Fonksiyon içerisinde float dizisini dönüyor önceki veri sonrakinden büyükse yer değiştiriyor. Floatla eş zamanlı aynı indexleri paylaştıkları için kolaylıkla string dizisinde de yer değişikliği yapıyor. Böylelikle satırlar ortalamalarına göre sıralandı. YoneticiListesi kurucusu parametre olarak dosyanın adresini(Dosyada yapılan işlemlere ve fonksiyonlara erişebilmek için) ve satir sayısını(fonksiyonlarda kullanılıyor) alıyor. YoneticiListesi kurucusunda bir SatirListesi'leri tutan SatirListesi Listesi oluşturdum(SatirListesi**). YoneticiListesi ekle komutu ile hem YDugum nesnelerini oluşturdum oluştururken SatirListelerine satırları atadım. Böylelikle tek fonksiyonda hem SatirListelerine verileri atadım hem de YDugum'lere SatirListelerini atadım. Test amaçlı main.cpp içinde YoneticiListesi nesnesi oluşturarak ilk düğüm adresini yazdırarak çalıştığını kontrol ettim. Tabii oldukça hata aldım ve hataları düzelttim. Hataları düzeltmekte dünya kadar süre aldı tabii.. Destekleyici ya da aracı fonksiyonları atlarsak sıra Yazdır fonksiyonuna geldi. Yazdır fonksiyonu hem ekrana uygulamayı yazdırıyor hem de komutlarımızı girerek uygulamayı kullanmamızı sağlıyor. İçeriğine gelirsek uygulama temel olarak 2 bölüme ayrılıyor. Biri YöneticiListesi elemanlarını sayfa başına 8er olmak üzere ekrana yazdıran komut, diğeri ise seçili YoneticiListesi dugumune ait SatirListesi(YDugum) dugumlerini altına yazdırıyor. Daha sonra ödevde arz edildiği üzere belli komutlar ve belli işlevleri var. Sıra bunları eklemeydi. Ancak ödevde en zorlandığım kısım burası olabilir. Oldukça komplike kodlar oluşturdum daha sonra toplayarak bazılarını değiştirerek, bu sırada aldığım hataları ve geçen zamanı söylemiyorum bile, oldukça zorlanarak sade ve çalışır bir program elde ettim. Son olarak parametrelerin tamamını referans tanımlamamıştım. Gerekli olanları referans

tanımladım. Program bitmiş sayılırdı. Biraz programı zorlamak istiyordum. Hatalarını keşfetmek için uğraştım. Bulduğum hatalardan bazıları: ilk son yazıları sayfada buglanmaya neden oluyordu sayfa yazdırılmıyordu, son sayfadaki ydugum sayısı kadar ydugum silince son sayfaya geçilebiliyor program çöküyordu, bütün düğümleri silince program çöküyordu, 'k' komutu yanlış çalışıyordu vs....

Bu hataları inat ettim çözdüm. Programda açık bırakmamak için kontroller yaptım. Kendim bir hata bulamadım daha fazla. Elimden geleni yaptım. Başarılı bir program olduğunu düşünüyorum ama tabii ki bu benim düşüncem. Umarım bir açığı kalmamıştır programın.

Zorlandıklarım: En çok özellikle yazdır fonksiyonundaki komplike algoritmalar ve vurgulayarak söylemek istiyorum ufacık ama çok büyük Hatalar beni çok zorladı. Şunu da ekleyebilirim, Bağlı Listeler konusunda hep zorlanmışımdır ancak bu proje Bağlı Listeleri kafamda daha anlaşılır kıldı. Pardon düzeltmek istiyorum en çok vs code'ta cpp'lerin .\include\ içerisini görmemeleri yüzünden en fazla burada çile çektim. C++ 1696 Error verip duruyordu #include alamayınca kalıtım yapamaz kod yazarken çok zorlanır oldum. Çözümü şöyle buldum: include'taki header dosyaları src içerisine kopyaladım yaptığım değişiklikleri src içindekilerle yaptım çünkü kod yazarken anlık olarak cpp orayı görebiliyordu. Make ile çalıştıracağım zaman include içindekileri silerek src'deki header dosyalarını include'a kopyalıyordum. Öyle sürekli güncelleyerek çalıştım. Çözümü biliyorsanız mail yoluyla yardımcı olabilirsiniz sevinirim.

Eksik bıraktığım bir yer olduğunu düşünmüyorum. Çok fazla kontrol yaptım ama fark etmediğim hatam olabilir sizin takdirinize kalmış.

Umut Can ÇAĞATAY

2C G201210029