



SAKARYA
ÜNİVERSİTESİ

BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ

BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

VERİ YAPILARI ÖDEVİ

Öğrenci Adı: Mehmet

Öğrenci Soyadı: ATAŞ

Öğrenci Numarası: G201210382

Grup: 2/A

Öğretim Görevlisi: Dr. Muhammed Fatih ADAK

Ödev için ilk olarak "Node" adında düğüm oluşturan bir sınıf oluşturdum. Bu sınıfı oluştururken şablon veri yapısından faydalanmadım ve oluşturduğum sınıfı "Node.hpp" başlık dosyası ve "Node.cpp" kaynak dosyasına ayırdım. Bu sınıfın içerisinde düğümün verisini tutacak olan "string data" değişkenini, düğümün ilerisini ve gerisini gösterecek pointer'ları tanımladım. Ardından sınıf ismi ile aynı olan "Node" kurucu fonksiyonunu yazdım. Daha sonra çift yönlü bağlı liste fonksiyonlarını sağlayacak olan "DoublyLinkedList" sınıfını oluşturdum ve aynı şekilde bu sınıfı da kaynak ve başlık dosyalarına ayırdım.

"DoublyLinkedList" sınıfına kurucu fonksiyon, eleman ekleme işlemini gerçekleştiren "insert" fonksiyonu, eleman silme işlemini gerçekleştirecek "removeAt" fonksiyonu, listeyi ekrana yazdırmayı sağlayan fonksiyonu ve sınıfa ait yıkıcı fonksiyonu yazdım. Aynı zamanda içerisine girilen index'in bir öncesini gösteren "FindPreviosNode" isimli bir fonksiyon da tanımladım.

"insert" fonksiyonunda ilk olarak girilen index değerinin sıfırdan küçük olup olmadığını sorguladım. Eğer girilen değer sıfırdan küçük ise hata fırlatılmasını sağladım. Ardından girilen index'e göre ekleme işlemini yapacak if-else if -else yapılarını oluşturdum. Bu yapılara girilen index'e ve iki yönlü bağlı listenin boyutuna göre ekleme işlemlerini gerçekleştirecek kodları yazdım.

-Ekleme işlemi 3 şekilde gerçekleştirilmekte:

- 1-Girilen index her ne olursa olsun listenin boyutu sıfırsa, eleman listenin başına eklenir.
- 2-Listenin boyutu sıfır değilse ve veri eklenmek istenen index listenin boyutundan büyükse; listenin en sonuna düğüm eklenir ve veri düğüme yazdırılır.
- 3-Diğer durumlarda ise veri eklenmek istenen index'e veri eklenir. Veriler direkt istenilen indexe, araya düğüm ekleyerek eklenmez. Düğüm listenin sonuna eklenir, veriler kaydırılır ve veri, eklenmek istenilen index'teki düğüme yazdırılır.

"removeAt" " fonksiyonunda ilk olarak girilen index değerinin sıfırdan küçük olup olmadığını sorguladım. Eğer girilen değer sıfırdan küçük ise hata fırlatılmasını sağladım. Ardından "del" isminde, silinecek düğümü göstermemi sağlayacak olan bir pointer tanımladım. Bu fonksiyon içerisinde de if-else yapısı kullanarak verinin hangi şekilde silineceğine karar verecek bir yapı oluşturdum. Girilen index'e ve listenin boyutuna göre silme işleminin yapılmasını sağladım.

Silme işleminde bazı durumlarda direkt istenilen index'teki düğüm silinmez. Girilen index listenin boyutundan büyükse en sondaki düğüm silinir. Listenin boyutuna göre silinmek istenilen index'in sağından başlayarak veriler bir sol tarafa kopyalanır ve en sondaki düğüm silinir.

Bu işlemlerden sonra dosyadan veri okuma işlemini gerçekleştirmek için "DosyaOku" adında bir sınıf tanımladım. Bu sınıfı da "DosyaOku.hpp" başlık dosyasına ve "DosyaOku.cpp" kaynak dosyasına ayırdım. "DosyaOku" sınıfı içerisinde bulunan "dosyaOku" fonksiyonu dosyadan veri okunmasını sağlar ve dosyadan gelen komuta göre ekleme veya silme işlemini gerçekleştirir. Dosyadan okunan komutlar bittikten sonra listenin son hali ekrana yazdırılır ve uygulama son bulur.

Ödevi genel olarak derste öğretilen bilgilerle adım adım, hatalar alıp, aldığım hataları düzelterek tamamladım. Bu ödev iki yönlü bağlı liste yapısını (genel olarak liste yapılarını) ve pointer kavramını daha iyi kavramamı sağladı.