

T.C.
SAKARYA ÜNİVERSİTESİ
BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

Ders : **Veri Yapıları**
Dönem : **2021-2022 Güz Dönemi**

Ad ve Soyad : **Muhammet Kemal Güvenç**

Okul No : **B181210076**

Konu : **1.Ödev**

Şube : **1.Öğretim A Grubu**

Yazılan kodda değişkenler, metot adları ve sınıf adları kodun evrensel olması için İngilizce olarak yazılmıştır. Fakat kodları size en iyi şekilde anlatabilmek için yorum satırları Türkçe olarak yazılmıştır. Değişkenler küçük harflerle yazılmış eğer birden fazla kelimeden oluşuyorsa araya _ işareti konularak adlandırılmıştır. Sınıf ve fonksiyon adları ise bu kurala ek olarak baş harfleri büyük olarak adlandırılmıştır.

Projede Node, Iterator ve List olmak üzere toplam 3 sınıf bulunmaktadır. Bütün sınıflarda veriler, kodun güvenliği için get ve set metotları üzerinden alınıp, verilmiştir. Ayrıca veriler üzerinde değişikliğe gitmeyen fonksiyonlar const olarak belirtilmiştir. Sınıflar, başlık dosyası ve kaynak dosyası olmak üzere 2 dosyaya yazılmıştır. Başlık dosyalarında sınıfların üye değişkenleri ve metot bildirimleri, kaynak dosyalarında ise metotların gövdeleri bulunmaktadır. Kod yazılırken elden geldiğince gereksiz bellek tüketiminden kaçınılmıştır.

Node sınıfı, kendi içinde tuttuğu eleman, ilerisinde ve gerisinde olan düğümlerin adresleri olmak üzere 3 üye değişkene sahiptir. Bir de her bir elemana ait get ve set metotları (geride olan düğüm adresine ait set metodu hariç) olmak üzere 5 tane metot bulundurmaktadır. Tabii, bunlara ek olarak bir de yapıcı metodu vardır. Düğüm sınıfından bir nesne oluşturulurken düğümün içine konacak eleman ve gerisinde olan düğümün adresinin verilmesi şart koşulmuştur. Düğümün ilerisi ise otomatik olarak NULL'a eşitlenir.

Iterator sınıfı gösterdiği düğümün adresini tutacak bir gösterici ve üzerinde bulunduğu düğümün indeks bilgisine sahiptir. Bir Iterator oluşturulurken bağlanacağı bir düğüm adresi ve onun indeksinin verilmesi şart koşulmuştur. Buna karşılık yapıcı metodu dahil olmak üzere 6 adet metodu bulunmaktadır. Bunlardan 2'si gösterilen düğümün adresi ve indeksinin get metotları diğer 2'si ise Iterator'ün bağlı olduğu düğümden ileri ve geri gitmesini sağlamaktadır. Diğerleri ise Iterator'ün ileriye gidip gidemeyeceğini kontrol eder.

List sınıfında 2 tane üye değişkeni bulunmaktadır. Bunlar listenin başını işaret eden gösterici ile listenin boyutunu tutan değişkendir. Bu sınıftan yeni nesne oluşturulduğunda göstericiye boş bir düğüm, boyuta 0 atanır. Yapıcı ve yıkıcı fonksiyonlar dışında 2 adet metodu bulunmaktadır. Bunlarda biri listenin belirli bir indeksine ekleme yapmamızı sağlayan Add fonksiyonu diğeri ise listenin belirli bir indeksindeki veriyi silmemizi sağlayan Remove fonksiyonudur. Bunların dışında "<<" operatörü listenin gösterilebilmesi için aşırı yüklenmiştir. Add ve Remove fonksiyonlarına şöyle bir bakmak gerekirse:

Add: İlk önce gönderilen indeks kontrol edilir. Eğer indeks sıfırdan küçük veya listenin boyutundan büyük ise indeks sınıfın boyutun eşitlenir. Böylece veri listenin sonuna eklenir. Bundan sonra listenin başına konumlanacak şekilde bir Iterator oluşturulur. Sonra bu Iterator listenin sonuna götürülür. Iterator listenin sonuna geldikten sonra bir düğüm oluşturulur ve bu düğüm listenin sonuna bağlanır ve Iterator'de o düğüme ilerletilir. Bunlardan sonra Iterator verinin ekleneceği düğüme götürülür fakat bu yapılırken bir yandan da düğümler arasında veriler kaydırılır. En sonda ise Iterator verinin ekleneceği düğüme gelir ve eklenecek veriyi oraya kaydeder.

Remove: İlk önce gönderilen indeks kontrol edilir. Eğer indeks sıfırdan küçük veya listenin boyutundan büyük ise indeks sınıfın boyutun eşitlenir. Böylece veri listenin sonundan silinir. Bundan sonra listenin başına konumlanacak şekilde bir Iterator oluşturulur. Sonra bu Iterator

belirtilen indeksteki düğüme götürülür. Bundan sonra ise Iterator düğümün sonuna götürülür ancak bunu yaparken bir yandan düğümler arasında veriler kaydırılır. Listenin sonuna gelen Iterator bir adım geriletilir ve ilerideki düğüm silinir.

Ana programın çalışması ise şu şekildedir: İlk önce Veri.txt dosyasını açmaya çalışır. Eğer açamaz ise "Veri.txt dosyasi bulunamadi. Veri.txt dosyasının "makefile" dosyasinin yanında oldugundan emin olun. Program kapatiliyor...." uyarısı verir ve kapanır. Dosya sıkıntısız bir şekilde açıldıktan sonra Veri.txt dosyasından komutlar satır, satır okunmaya başlar. Komutun ilk harfine göre listeye ekleme veya listeden silme yapılır. İndeks bilgisi komutta "(" ile "#" işaretleri arasında kalan bölgeden alınır. Eklenecek veri ise "#" ile ")" işaretleri arasında kalan bölgeden alınır. Dosya tamamen okunup ve tüm işlemler yapıldıktan sonra dosya kapatılır. Sonra tüm bu işlemler sonucu oluşan liste gösterilir.