

# BSM207 VERİ YAPILARI

## ÖDEV 2 RAPORU

Veri yapılarının ikinci ödevinde bizden, C++ ile bir metin dosyasını açıp içerisindeki metni kelime kelime okuyarak her bir kelimeyi oluşturduğumuz çift yönlü bağlı listenin bir düğümüne eklememiz istendi. Bu ekleme işlemini yaparken tekrar eden kelimeler olduğu durumda, tekrar eden kelimenin kaç düğüm öncesinde bulunduğu bilgisini tekrarlı kelimeyi içeren düğüme girmemiz gerekiyordu.

Çift yönlü bağlı liste sınıfında, temel get, set ve kurucu metotların dışında listenin boş olup olmadığını kontrol eden bir **isEmpty** metodu, listenin boyutunu döndüren **Count** metodu, listenin sonuna düğüm eklemek ve eklenecek kelimenin tekrarlı olup olmadığını kontrol eden **add** metodu, yazdırma işlemi için **<<** operatörünün overload işlemi ve heap bellek bölgesinde oluşan tüm nesnelerin geri iade edilmesinin yapıldığı **clear** metodu bulunur.

Düğüm sınıfında sadece temel alanlar, bu alanlar için get ve set metotları ve kurucu metot tanımlamaları bulunur.

Iterator sınıfında, temel get, set ve kurucu metotların haricinde iteratörün hareket edip edemeyeceğini döndüren **canMove** metodu, iteratörün bir sonraki düğüme ilerleten **next** metodu ve iteratörün o anda gösterdiği düğümün elemanını döndüren **getCurrentEleman** metodu bulunur.

Dosya sınıfında, parametre olarak verilen dosya yolundaki dosyayı açan **DosyaAc** metodu ve açılan dosyadaki metni parametre olarak verilen çift yönlü bağlı listeye kelime kelime aktaran **ListeyeAktar** metodu bulunur.

Bu ödevi yaparken çok zorlandığım ve beni yavaşlatan bir sıkıntıyla karşılaşmadım. Dosya okuma işlemlerini nasıl yapabileceğimle ilgili biraz Google araştırmasıyla hızlıca nasıl yapıldığını öğrendikten sonra bunu kendi koduma uyarladım. Derslerde zaten çift yönlü bağlı listenin nasıl yapıldığını detaylıca öğrenmiştik. Derste yazdığım kodlar üzerinde biraz değişiklikler yaparak ödevde bizden istenenleri karşılayacak şekilde kodları yeniden düzenledim. Özellikle add metodunu sadece listenin sonuna eleman ekleyeceği şekilde düzenledim. Bunun yanında add metodu içerisinde listenin sonuna kadar her düğümden geçerek giden bir iteratör vardı ve ben de bu iteratörü her düğümden

geçerken bir de geçtiği düğümün içeriğini yeni eklenecek olan içerik ile karşılaştırıp tekrarlı değer eklenip eklenmediğini kontrol eden bir yapıyabüründürdüm.

Bu ödevde makefile oluşturma ve farklı dosyalarla çalışma konusunda hiç sıkıntı yaşamadım. Önceki ödevden en büyük kazancım bu oldu.

Bu ödevden kişisel olarak en büyük kazancım pointerları kullanma becerisi oldu.

Programı en son haline getirdikten sonra birkaç farklı text dosyasıyla denemeler yaparak test ettim. Benim görebildiğim herhangi bir hatayla karşılaşmadım. Program ödevde istenilen bütün işlemleri yapabilecek seviyededir. Programa ödev dosyasında verilen metinlerin verilir çalıştırıldığı ekran görüntüsü ve nispeten uzun bir lorem ipsum metni verildiğindeki ekran görüntüsü aşağıda eklenmiştir. Programa 100 bin kelimelik bir metin verip 5dk bekleme sonrasında hala işleyemediğini görüp kapattım. Bunun sebebinin eklenen her bir kelime için listenin sonuna kadar iteratör ile gitmesi olduğunu düşünüyorum. 50 bin kelimelik bir metni 1-2 dakika içerisinde işleyip ekrana basabiliyor.

Bütün kodlar Notepad++ üzerinde yazılıp MinGW derleyicisi ile derlendi.

```
C:\Windows\System32\cmd.exe - mingw32-make

C:\Users\Arif\Desktop\VeriYapıları0dev2\G171210009_VeriYapilari_Odev2>mingw32-make
g++ -I ./include/ -o ./lib/Dugum.o -c ./src/Dugum.cpp
g++ -I ./include/ -o ./lib/Iterator.o -c ./src/Iterator.cpp
g++ -I ./include/ -o ./lib/DoublyLinkedList.o -c ./src/DoublyLinkedList.cpp
g++ -I ./include/ -o ./lib/Dosya.o -c ./src/Dosya.cpp
g++ -I ./include/ -o ./bin/test ./lib/Iterator.o ./lib/Dugum.o ./lib/DoublyLinkedList.o ./lib/Dosya.o ./src/test.cpp
./bin/test
NULL<-|Sakarya - 0|<->|Universitesi - 0|<->| - 2|<->|kampusu - 0|<->|Esentepe - 0|<->| - 2|<->| - 6|<->|NULL
Toplam Kazanc: 18

Press any key to continue . . .
```

