

VERİ YAPILARI 2019-2020 1. ÖDEV RAPORU

Yazdığım program bir C++ konsol uygulamasıdır. Program çalıştığında içinde belirli bir aralıkta pozitif integer değerlerin olduğu bir text dosyasını ifstream ile okur. getline() metodu ile integer değerlerin satır satır okunup, çift yönlü dairesel bağlı liste nesnesine atanması sağlanır. Değerler başlangıçta string veri tipinde olduğu için her sayının, okunduğunda öncelikle istream ile integer veri tipine dönüştürülmesi sağlanır.

Programın algoritması ile ilgili olarak, öncelikle başlangıçta EBOB değeri 0 kabul edilir. Bu kabul, ilk değer okunduğunda da devam eder çünkü henüz kıyaslama yapılacak ikinci bir değer bulunmamaktadır. İkinci değer ve daha fazla değer okundukça EBOB değerleri de değişmektedir.

Node sınıfı düğüm sınıfıdır. .hpp başlık dosyası ve .cpp kaynak dosyası bulunmaktadır.

Iterator sınıfından türetilen nesnenin görevi çift yönlü dairesel bağlı listenin indisleri arasında kendisine verilen metotlar doğrultusunda iki yönlü olarak gezme. Iterator sınıfının .cpp kaynak dosyası ve .hpp başlık dosyası bulunmaktadır. .cpp dosyası src klasöründe, .hpp dosyası include klasöründe bulunur. .hpp başlık dosyasında metot gövdeleri bulunmaksızın metotların imzaları yer alırken .cpp klasöründe bu metotların gövdeleri yer almaktadır. Iterator sınıfı aşağıdaki metotları içerir.

```
bool IsEndNode();  
  
void StepNext();  
  
void StepBack();  
  
const int& GetCurrentData() const;
```

StepNext() ve StepBack() iteratörün düğümler arasında gezmesini sağlar.

StepNext(), current = current->next; ile ileri gitmeyi sağlarken ;

StepBack(), current = current->prev; ile geri gitmeyi sağlar.

GetCurrentData(), iteratörün üzerinde bulunduğu düğümdeki değeri döndürmektir.

CircularDoublyLinkedList sınıfı çift yönlü dairesel bağlı liste sınıfıdır. .cpp kaynak dosyası ve .hpp başlık dosyası gerekli klasörlerde bulunmaktadır. Başlık dosyasında sadece tanımlar yapılmıştır. Tüm metot gövdeleri kaynak dosyada bulunmaktadır Bu sınıf

```
Iterator IterateFromFirstNode() const; IterateFromPrevIndex(int index) const;  
  
void UpdateLastNode(); CircularDoublyLinkedList();  
  
IsEmpty() const; Count() const; const int& First() const; const int& Last() const;  
  
void Add(const int& obj); void Insert(int index, const int& obj);
```

void Remove(const int& obj); void RemoveAt(int index); int IndexOf(const int& obj) const;
void Clear();

metotlarını içeren birçok işlevi olan oldukça kapsamlı bir sınıftır. Bazı önemli metotların işlevleri şu şekildedir:

Iterator IterateFromFirstNode() const; → İlk düğümden başlayarak iterasyon.

IterateFromPrevIndex(int index) const; → parametre olarak aldığı indis değerinin bir öncesinden başlayarak iterasyon.

void UpdateLastNode(); → Son düğümü günceller, lastNode->next = head->next atamasıyla daireselliği sağlar.

IsEmpty() const; → Listenin boş-dolu olmasını kontrol eder.

Count() const; → Size değerini döndürerek listedeki eleman sayısını verir.

const int& First() const; const int& Last() const; → İlk ve son elemanları döndürür.

void Add(const int& obj); void Insert(int index, const int& obj); → Add ile sona eleman eklenir, insert ile ise belirli bir indise eleman eklenir.

void Remove(const int& obj); void RemoveAt(int index); → Remove ile sondan eleman silinir, RemoveAt ile ise belirli bir indisteki eleman silinir.

Ödevde, okunan sayıların EBOBları hesaplanarak en son EBOBun , o ana kadarki en büyük EBOB değerinden büyük olması, küçük olması veya o ana kadarki en büyük EBOB değerine eşit olması durumlarında yapılacaklar, iki sayının modu, modun iteratörün üzerinde bulunduğu indisten farkı alındığında 0'dan büyük, küçük veya eşit olması durumları, bu durumlara göre yapılması istenenlerin ödev dökümanında istendiği ve örneklendiği şekilde yapılması sağlanacak şekilde gerekli veri yapısı ve algoritma tasarlandı. Program sonunda elemanların silinmesini sağlayan Clear() fonksiyonu çağrıldı. Program dökümanda istendiği gibi MinGW'de derlenecek şekilde hazırlandı. Program çalıştırıldığında yine dökümanda istendiği üzere hesaplamalar gösterilmedi. Program çalıştırıldığında şifre değerlerini satır satır ekrana basıp programdan çıkış yapmaktadır.

Bu ödev tek yönlü bağlı liste, çift yönlü bağlı liste, tek yönlü dairesel bağlı liste ve çift yönlü dairesel bağlı listeden oluşan tüm bağlı liste çeşitlerinin neler olduğunu, nasıl ve ne amaçla kullanıldıklarını ödevi hazırlamam süresince yaptığım araştırmalarla daha iyi öğrenmemi ve kavramamı sağladı.

Ödevi hazırlarken yararlandığım kaynaklar sanal ders konu anlatımı, ve C/C++ İle Veri Yapıları ve Çözümlü Uygulamalar kitabı oldu. Kitap, düğüm oluşturma , liste gezici oluşturma , bağlı liste oluşturma hususlarını , konu anlatımı ve çözümlü örnekleri ile anlamamı sağlayarak ödevimde yardımcı oldu.

Ödevi hazırlarken en çok zorlandığım kısım, üzerinde fazla pratiğimin olmadığı ancak bu dersin en önemli mefhumu olan pointer kullanımı idi.