

İSİM: Emirhan

SOYİSİM: KAYA

NUMARA: G201210010

ÖDEV NO: 1

GRUP: 2A

İstenilen

Veri.txt adında bir dosya içerisinde verilenlere göre 2 yönlü bağlı liste içerisinde dizi mantığı ile eleman eklemek ve silmek. Eleman eklerken kapasitenin dışında bir indekse eleman eklenilmeye çalışılırsa listenin sonuna eleman eklemek. Eleman silerken kapasitenin dışında bir indeksten eleman silinmeye çalışılırsa listenin sonundan eleman silmek.

Öğrendiklerim

2 yönlü bağlı listenin yapısı, kodlanış şekli ve istenilen durumlara göre 2 yönlü bağlı listeyi projelerde kullanabilmeyi öğrendim. Bağlı listeler ve dizilerin arasında ne gibi farklar olduğunu anladım. Örnek olarak dizide eleman eklerken elemandan sonraki bütün elemanlar kendisinden bir sonraki indekse kopyalanırken, 2 yönlü bağlı listede eleman eklenecek indeksteki düğüm ile bir önceki düğüm arasına pointerlar ile yerleştiriliyor. Herhangi bir kopyalama işlemine gerek kalmıyor.

Ödevde Yaptıklarım

Node adında bir düğüm sınıfı oluşturdum. İçerisinde string isim tutacak item ve next, prev düğüm pointerları bulunmakta.

Iterator adında Node dosyasını include eden iteratör sınıfı oluşturdum. Bu sınıf benim bağlı liste içerisinde gezinmemi sağlayacak olup kendi içerisinde current adında bir düğüm pointerı bulundurmakta. Bu pointer iteratörün bulunduğu indeksteki düğümü göstermekle görevlidir. Aynı olarak Iterator sınıfının içerisinde next ve prev adında pointerlar yardımı ile iteratörün ileri ve geri indekslere gitmesini sağlayan fonksiyonlar bulunmaktadır.

LinkedList adında Iterator dosyasını include eden 2 yönlü bağlı liste sınıfı oluşturdum içerisinde projede istenilenlere göre kullanabileceğim fonksiyonların kalacağı şekilde doldurdum. Sınıf içerisinde private olarak bağlı listenin ilk elemanını gösteren head adında düğüm pointerı, listenin eleman sayısını tutan integer size değişkeni, bulmak istediğimiz indeksteki düğümden bir önceki düğümü gösteren iteratörü döndüren FindPrevByPosition adında fonksiyon bulunmaktadır. Public olarak ise listenin sonuna eleman ekleyen add fonksiyonu, listede herhangi bir indekse **bağlı liste mantığı ile** eleman ekleyen insert fonksiyonu, listede herhangi bir indeksteki düğümü **bağlı liste mantığı ile** silen removeAt fonksiyonu, listedeki herhangi bir indekse eleman eklemek için ilk önce son indekse insert fonksiyonu ile eleman ekleyen sonrasında ise eklemek istediği indekse kadar bir önceki indeksteki düğümün itemını bulunduğu indeksteki düğümün itemına kopyalama işlemi yapan ve indekse geldikten sonra eklenecek ismi indeksteki düğümün itemına atayan **Ekle** fonksiyonu, elemanın silinmek istediği indeksten başlayıp, indeksteki düğümden sonraki düğümün itemını bulunduğu indeksteki düğümün itemına kopyalayarak son indekse kadar gelen ve sonunda removeAt fonksiyonunu kullanarak bağlı listenin son elemanını silen **Sil** fonksiyonu, bağlı listedeki elemanları ödevde istenilen şekilde ekrana yazdıran overload edilmiş << operatörü ve listedeki tüm elemanları silen clear fonksiyonu bulunmaktadır.

Test.cpp adında LinkedList dosyasını include eden bir main dosyası oluşturdum ve dosya okuma işlemlerini bu dosyada yaptırđım. İlk önce isimler adında LinkedList pointeri oluşturdum ve new anahtarı ile bağılı listeyi heap bölgede oluşturdum. Sonrasında Veri.txt dosyası içerisindeki her bir satırı okuyup satırlardaki 0. İndeksin E veya S olmasına bağılı 2 farklı durum oluşturdum. E olması durumuna göre aç parantezinden sonra kareye kadar indeks tanımı kareden sonraki elemandan kapa paranteze kadar da isim tanımı yaptırđım. S olması durumuna göre ise de aç parantezden sonraki elemandan kapa paranteze kadar alıp indeks tanımı yaptırđım. Bu işlemler için substr fonksiyonunu kullandım. E olması durumu için bulduğıım isim ve indeksi LinkedList->Ekle(isim,index) şeklinde kullandım. S olması durumu için ise bulduğıım indeksi LinkedList->Sil(index) şeklinde kullandım. Veri.txt dosyasının tamamı okunduktan sonra ise << operatörü ile bağılı listenin içerisindeki tüm düğümleri yazdırđım. En sonunda ise çöp kalmaması için delete operatörü ile isimler pointerimin gösterdiği bağılı listeyi heap bölgeden sildim ve programı bitirdim.

Eksik Bıraktıklarım

Ödevde herhangi bir eksik bıraktığıım yer bulunmamaktadır.

Zorlandığıım Kısımlar

İki yönlü bağılı listeye eleman eklerken bir dizi gibi davranmasını sağlamakta zorlandım. Bu noktada asıl zorlandığıım kısım elemanları kopyalama işlemiydi. Çünkü ilk önce düğümün içerisindeki itemı kopyalamak yerine direk düğümü kopyalamaya çalıştım. Böyle olunca da kopyaladığıım düğüm yazdırma fonksiyonunda sonsuza kadar yazdırılıyordu. Hal böyle olunca ilk başta ne olduğunu anlamadım. Sonrasında kodumu sırayla çizerek kontrol ettiğimde kopyalama işleminde kopyaladığıım düğümün kendisini gösterdiğini anladım. Bu yüzden kopyalanan itemdan sonra 2 yönlü bağılı liste tek yönlü çembersel bağılı liste gibi davranmaya başladığıından yazdırma işlemi kopyalanan itema geldikten sonra sonsuza kadar itemın kendisini yazdırmaya devam ediyordu. Bu sorunu düğümü direk kopyalamak yerine düğümlerin içerisindeki itemları kopyalayarak çözdüm.

```
Microsoft Windows [Version 10.0.19042.1348]
(c) Microsoft Corporation. Tüm hakları saklıdır.

C:\Users\babai\Desktop\veriYapilari_ODEV1>mingw32-make
g++ -I ./include/ -o ./lib/Node.o -c ./src/Node.cpp
g++ -I ./include/ -o ./lib/Iterator.o -c ./src/Iterator.cpp
g++ -I ./include/ -o ./lib/LinkedList.o -c ./src/LinkedList.cpp
g++ -I ./include/ -o ./bin/Test ./lib/Node.o ./lib/Iterator.o ./lib/LinkedList.o ./src/Test.cpp
./bin/Test
E(2#Mehmet Can)
E(25#Ali)
E(0#Kenan)
E(2#Ersin)
E(3#Mazhar)
S(3)
S(49)
Kenan<-->Mehmet Can<-->Ersin

Programı kapatmak için herhangi bir tusa basıp enter'a basınız.
```

Resim 1.0: Test.exe programından bir görüntü (Not: Programın son halinde sıralı işlemler ekrana yazılmayacaktır.)