Sait ORHAN

1030515650

# Veri Yapıları ve Algoritmalar Dersi Ödev 1 Raporu

# Veriler Sınıfı Metotları

## void BolumBelirle(void);

Öğrencinin okuduğu bölümü tutmak için kullanılan bolum[15] değişkeninin değeri öğrencinin girmiş olduğu numarasına göre otomatik olarak sistem tarafından belirlenmektedir. Bunun için beş basamaklı olan öğrenci numarası analiz edilmektedir. Analiz işlemi şöyle işlemektedir: Öğrenci numarasının ilk iki rakamı okuduğu bölümün kodunu göstermektedir. Void BolumBelirle(void) metodu da girilen numaraya göre bölümü belirleyen metottur.

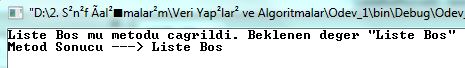
## void BolumYaz(void);

bolum[15] değişkeninin private olmasından dolayı öğrencinin bölümünü ekrana yazmak için kullanılmıştır.

# Liste Sınıfı Metotları

## bool ListeBosMu(void);

ListeSonu ve ListeBasi işaretçileri NULL değerine sahip iseler listede eleman yoktur ve true değerini döndürür.

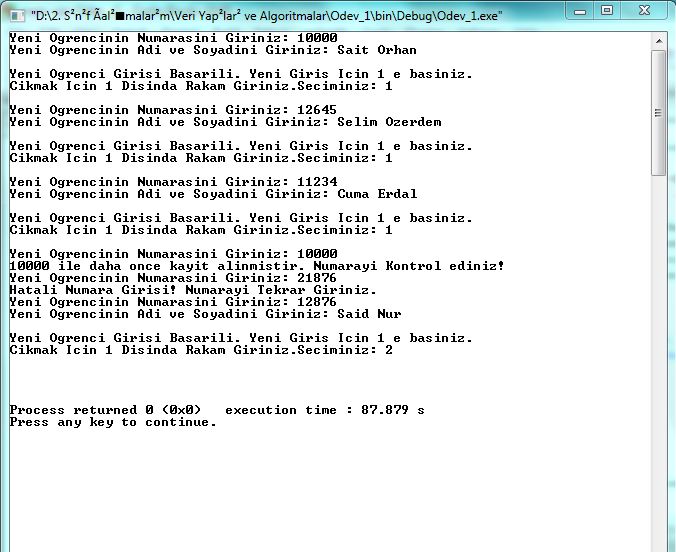


## void ListeyeEkle(void);

Listeye belirsiz sayıda elaman eklemek için kullanılmıştır. Çalışma mantığı:

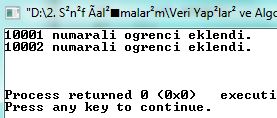
İlk adım olarak yeni bir alan tahsisi yaptıktan sonra eklenecek öğrencinin numarasını alır. Aldığı numarayı analiz ederek derse kayıt yapabilecek bir numara olup olmadığını test eder. Dersi alabilen öğrenci numaraları [10000, 19999] aralığındaki numaralardır. Daha sonra alınan numara bool OgrenciBul(const int ArananNumara) metoduna gönderir. Dönen değer true ise bu numaranın daha önce kayıt yaptığını ve tekrar kayıt yapamayacağı uyarısını verdikten sonra yeni numara girişi ister. Bu iki temel testi geçen numara aşağıdaki algoritmaya göre listeye eklenir. Daha sonra eklenecek eleman kopyalanır ve numaranın çift veya tek olmasına göre eklenecek elemanın kopyası Tekler listesi veya Çiftler listesine eklenmek üzere void ListeyeEkle(Veriler \*Eklenecek) metoduna gönderilir.

* Liste boş ise eklenecek eleman hem liste başı hem liste sonudur,
* Eklenecek numara liste başındaki numaradan küçükse liste başına eklenir ve yeni liste başıdır,
* Eklenecek numara liste sonundaki numaradan büyükse liste sonuna eklenir ve yeni liste sonudur,
* Diğer durumlarda araya eklenir, liste başından başlanarak eklenecek numara bakılan numaradan küçük oluncaya kadar liste taranır ve yeri bulunduktan sonra da araya eklenir.



## void ListeyeEkle(Veriler \*Eklenecek);

Listeye tek eleman eklemek için kullanılır. İlk adım olarak gelen öğrencinin numarasını analiz ederek kayıt yapabilen bir numara ise BolumBelirle(); metodu çağrılarak bölümü atanır. Eğer kayıt yapamayacak bir öğrenci ise “kayıt yapamaz” uyarısı vererek metodu sonlandırır. Ekleme işlemi daha önce açıklanan ekleme metoduyla aynı çalışır.



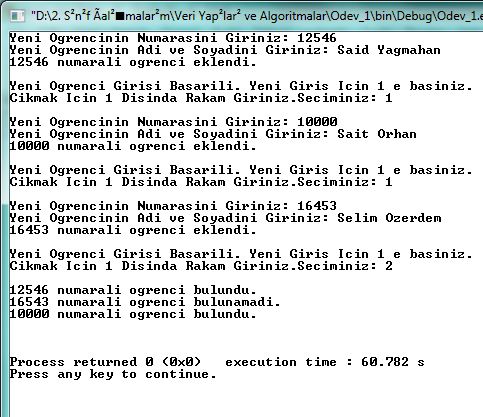
## bool OgrenciBul(const int ArananNumara);

Metot parametra olarak bir öğrenci numarası alıp bunu listede arar, bulması durumunda true aksi durumlarda false değerini döndürür. Metodun numara ararken kullandığı mantık:

* Liste boşsa false döndür,
* Aranan numara liste başından küçükse listede yoktur ve false döndür,
* Aranan numara liste sonundan büyükse listede yoktur ve false döndür,
* Aksi durumlarda: Listenin başını gösteren bir işaretçi al; bakılan numara aranan numaradan farklı ise işaretçiyi bir sonraki elemana kaydır. Bu işlem işaretçi NULL oluncaya veya numara aranan numaraya eşit oluncaya dek tekrarla. Döngüden çıkışta işaretçi NULL gösteriyorsa liste tamamen taranmıştır ve bulunamamıştır, false döndür; numara aranan numaraya eşit ise bulunmuştur true döndür.

|  |  |  |  |
| --- | --- | --- | --- |
| 10000 | Bilgisayar | Sait | Orhan |
| 12546 | Cevre | Said | Yagmahan |
| 16453 | Mekatronik | Selim | Ozedem |

Yukarıda ki listeye göre OgrenciBul metodu çalıştırıldı.

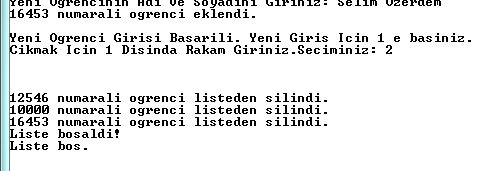


## void VeriSil(const int SilinecekNumara);

Listeden eleman silmek için kullanılır. Parametre olarak kaydı silinecek öğrencinin numarasını alır ve silme adımları için aşağıda ki işlemleri yapar;

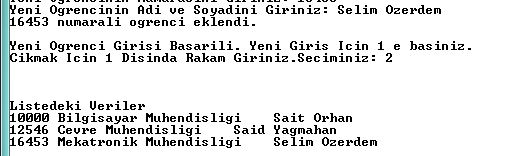
* Liste boş ise “Liste Boş” uyarısı ver ve metottan çık,
* Listede tek eleman varsa ve silinecek eleman ise liste başını ve liste sonunu NULL olarak ata,
* Silinecek eleman listenin ilk elemanı ise liste başını bir sonrakine kaydır,
* Silinecek eleman listenin son elemanı ise listenin sonunu bir öncekine kaydır,
* Yukarıdaki şartlardan hiç biri sağlamıyorsa listenin içinden arama yapmak için listenin başını gösteren bir işaretçi al, bu işaretçinin numarası silinecek numaraya eşit değilse işaretçiyi bir sonrakine kaydır. Bu döngüyü işaretçi NULL gösterinceye dek veya işaretçinin numarası silinecek numaraya eşit oluncaya dek tekrar et. İşaretçi NULL gösteriyorsa numara listede bulunamadı, işaretçi numarası silinecek numaraya eşit ise işaretçinin öncekinin sonrakini işaretçinin sonraki olarak değiştir, işaretçinin sonrakinin öncekini işaretçinin önceki olarak değiştir ve işaretçiyi sil.

\*Yukarıdaki tablo verileri girildikten sonra her biri için silme metodu çağrıldı.



## void ListeyiGoruntele(void);

Listeyi görüntülemek için kullanılır. Liste başını gösteren bir işaretçi alınır, işaretçinin verilerini ekrana yaz ve işaretçiyi bir sonrakine ilerlet. Bu döngü işaretçi NULL gösterinceye kadar devam eder.



## Veriler\* ListeBasiDondur() , Veriler\* ListeSonuDondur()

Liste sonu ve liste başı değişkenleri private olduğundan dolayı değişkenlerin başka değişkenlere ulaşım gerektiğinde döndürmek için kullanılır.

# Yığın Sınıfı Metotları

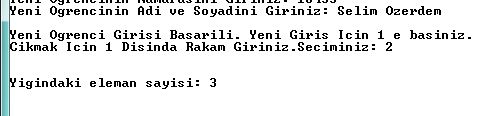
## bool YiginBosMu(void)

Yığının boş olup olmadığını sınar. Yığın Tepesi NULL gösteriyorsa yığın boştur, true döndür aksi halde false döndür.

## int ElemanSayisi(void);

Yığının eleman sayısını bulmak için kullanılır. Yığın tepesini gösteren bir işaretçi alınır ve işaretçi NULL gösterinceye kadar işaretçi bir alta kaydırılır. Her döngü adımı sonunda başlangıç değeri 0 olan eleman değişkenini bir arttır ve döngüden çıktıktan sonra eleman değişkenini döndür.

\*Daha önce kullanılan tabloda ki veriler girişi yapıldıktan sonra metot çağrıldı.



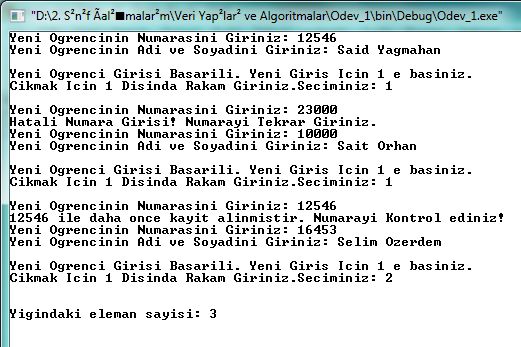
## bool YiginiTara(const int ArananNumara);

Yığını tepeden itibaren tarayarak aranan öğrencinin kayıt yapıp yapmadığını sınar. Çalışma mantığı Liste::OgrenciAra() metoduyla aynıdır.

## void TepeyeElemanEkle(void);

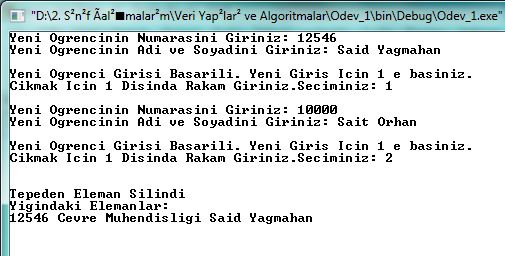
Yığına eleman eklemek için yeni alan tahsis edilir. Yeni elemanın numarası alındıktan sonra numara analiz edilerek kayıt olabilirliği sınanır daha sonra numara YiginiTara() metoduna gönderilerek daha önce kayıt yapmamış olduğu doğrulanır. Numara bu testlerden geçtikten sonra yeni elemanın sonrakine yığını tepesi değeri atanır ve yeni eleman da yığın tepesi olarak değiştirilir.

\*Yukarıdaki tablo verilerini yığına eklemek üzere metot çağrıldı.



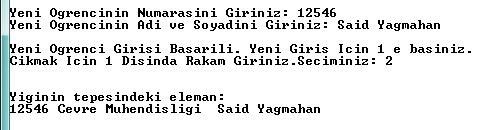
## void TepedenElemanSil(void);

Yığın tepesi NULL gösteriyorsa yığın boştur, değilse; yığın tepesini gösteren bir işaretçi alınır, yığın tepesini yığın tepesinin sonraki olarak atadıktan sonra eski yığın tepesini gösteren işaretçiyi sil.



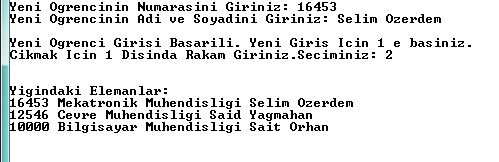
## void TepeElemaniniGoster(void);

Tepe elemanının bilgilerini yazar.



## void TumElemanlariYaz(void);

Yığının tepesini gösteren bir işaretçi alınır, bu işaretçinin bilgileri yazdırılır ve işaretçinin yeni değerine işaretçinin sonrakinin değeri atanır.



## YiginVerileri\* YiginTepesiDondur()

Yığın tepesinin private olmasından dolayı sınıf dışından yığın tepesi ihtiyacı olması durumunda döndürmek için kullanılır.

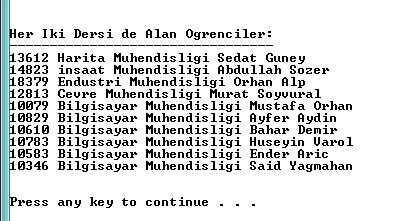
# Global Metotlar

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Numara | Bölümü | Adı Soyadı | V.Tabanı | Prog |
| 10346 | Bilgisayar Mühendisliği | Said Yağmahan | X | X |
| 10583 | Bilgisayar Mühendisliği | Ender Ariç | X | X |
| 10783 | Bilgisayar Mühendisliği | Hüseyin Varol | X | X |
| 10610 | Bilgisayar Mühendisliği | Bahar Demir | X | X |
| 10829 | Bilgisayar Mühendisliği | Ayfer Aydın | X | X |
| 10079 | Bilgisayar Mühendisliği | Mustafa Orhan | X | X |
| 11210 | Elektrik Mühendisliği | Zübeyir Dinç | X |  |
| 11812 | Elektrik Mühendisliği | Abdullah Kaya | X |  |
| 13709 | Harita Mühendisliği | A.Rahman Kaya | X |  |
| 18612 | Endüstri Mühendisliği | Adem Toy | X |  |
| 19312 | Yazılım Mühendisliği | Zekeriya Gündüz | X |  |
| 13513 | Harita Mühendisliği | Yunus Seven |  | X |
| 13110 | Harita Mühendisliği | Mustafa Saruhan |  | X |
| 14250 | İnşaat Mühendisliği | Hadi Kızmaz |  | X |
| 17003 | Biyomedikal Mühendisliği | Selim Özerdem |  | X |
| 16990 | Mekatronik Mühendisliği | Selman Özcan |  | X |
| 19112 | Yazılım Mühendisliği | Halime Gül |  | X |
| 15873 | Makina Mühendisliği | Mehmed Üney |  | X |
| 12813 | Çevre Mühendisliği | Murat Soyvural | X | X |
| 18379 | Endüstri Mühendisliği | Orhan Alp | X | X |
| 14823 | İnşaat Mühendisliği | Abdullah Sözer | X | X |
| 13612 | Harita Mühendisliği | Sedat Güney | X | X |

\*Global metotlarda kullanılan öğrenci listesi

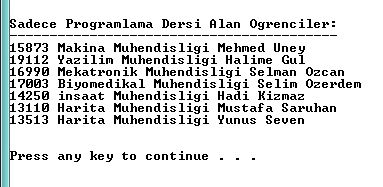
## void HerIkiDersiAlanlar(void);

Hem Programlama hem de Veri Tabanı derslerini alan öğrencileri yazdırmak için kullanılır. Çalışma mantığı: Öncelikle yığın tepesini ve liste başını gösteren iki işaretçi alınır. Yığın elemanlarını yazdırma metoduna benzer mantıkla yığındaki elemanı ekrana yazmadan önce Liste::OgrenciBul() metodu yardımıyla öğrencinin Veri Tabanı dersini de alıp almadığı sınanır. Metot true değeri döndürdü ise öğrenci her iki dersi de almaktadır, ekrana yaz.



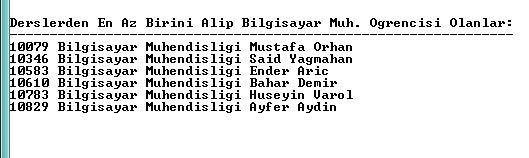
## void SadeceProgramlamaDersiAlanlar(void);

Çalışma mantığı: Öncelikle yığın tepesini ve liste başını gösteren iki işaretçi alınır. Yığın elemanlarını yazdırma metoduna benzer mantıkla yığındaki elemanı ekrana yazmadan önce Liste::OgrenciBul() metodu yardımıyla öğrencinin Veri Tabanı dersini de alıp almadığı sınanır. Metot false değeri döndürdü ise öğrenci sadece Programlama dersini almaktadır, ekrana yaz.



## void EnAzBiriniAlan\_Bilgisayar(void);

Öncelikle listede ki elemanları yazan metot mantığı ile liste ekrana yazdırılmaya çalışılır, ekrana yazdırılmaya çalışılan elemanın numarası 10 ile başlıyorsa Bilgisayar Mühendisliği öğrencisidir ekrana yaz. Daha sonra yığın ekrana yazdırılmaya çalışılır, bakılan elaman ekrana yazdırılmadan önce numara testine ek olarak öğrencinin iki defa yazdırılmasını engellemek amacıyla numarası listede aranır ve false değeri dönüyorsa ekrana yaz.



## Veri Tabanı Dersi Gruplara Ayrılmış Durumda:



