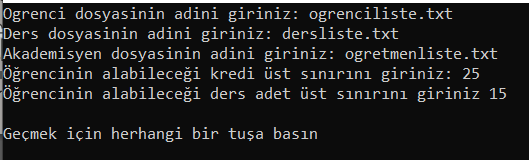
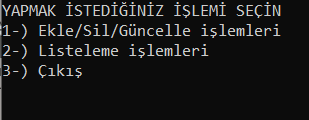
**İsmet Güngör – 19011100 – Grup2 – Yapısal Programlama**

1. **Menü Öncesi İşlemler**



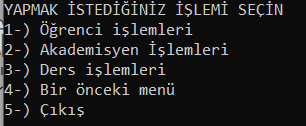
* Gerekli dosya isimleri ve istenen bilgiler girilerek ilerlenir.
* Dosya isimlerinde hata olması durumunda -1 hata koduyla program terk Edilir.
* Dosyalardaki bilgilerle birlikte öğrenci – ders kayıt dosyası oluşturulur.

1. **Ana Menü**

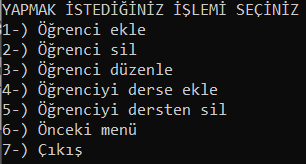


* Ana menü üzerinden yapılması istenen işlem seçilir.

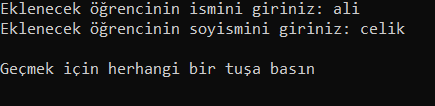
1. **Ekle/Sil/Güncelle İşlemleri Menüsü**



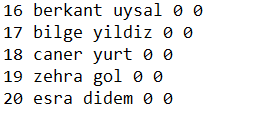
* Yapılması istenen işlem seçilerek devam edilir.
  1. **Öğrenci İşlemleri**

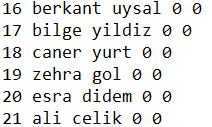


* Öğrenci işlemleri menüsü üzerinden yapılması istenen işlem seçilerek devam edilir.
  + 1. **Öğrenci Ekle**

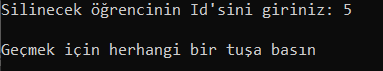


* Ekleme menüsünde eklenecek öğrencinin ismi ve soy ismi girilir.
* Öğrenci Id’si fonksiyon tarafından en güncel şekilde atanacaktır.
* Öğrencinin aldığı ders sayısı ve aldığı ders kredi 0 olarak atanacaktır.
* Birden fazla isme sahip kişiler isim1\_isim2 şeklinde tutulacaktır.

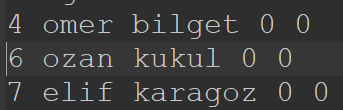


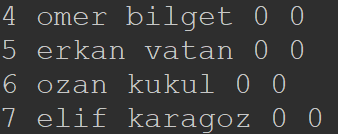


* + 1. **Öğrenci Sil**

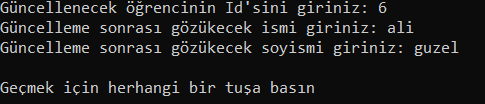


* Silme menüsünde öğrenci Id’si girilir.
* Öğrenci Id’si girildikten sonra fonksiyon arka planda daha önceden oluşturulmuş olan öğrenciDersKayıt dosyasında gerekli güncellemeler yapılarak silinen öğrenci hem öğrenci listesinden hem gerekli diğer dosyalardan silinir.
* Olmayan bir öğrenci Id’si girilirse hata alarak ana menüye yönlendiriliyorsunuz.

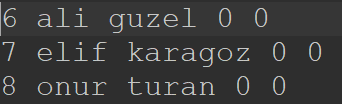


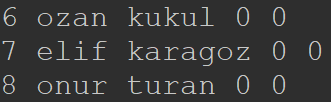


* + 1. **Öğrenci Düzenle**

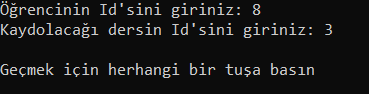


* Güncelleme menüsünde öncelikle öğrenci Id’si istenir.
* Öğrenci yoksa hata mesajı çıkarak sizi ana menüye yönlendirir.
* Doğru Id girildikten sonra güncelleme sonrası için gerekli bilgiler istenir.

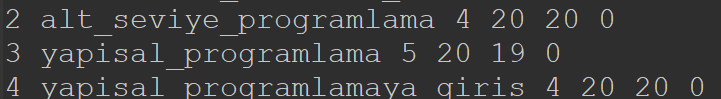


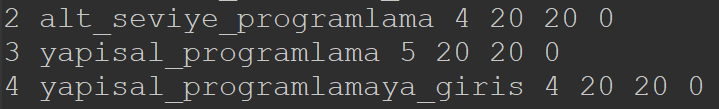


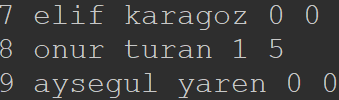
* + 1. **Öğrenciyi Derse Ekle**

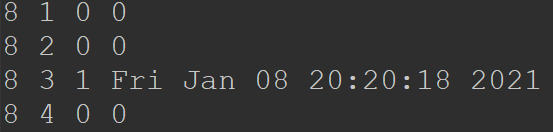


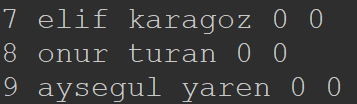
* Kullanıcıdan derse kaydedilecek öğrencinin ve kaydolacağı dersin Id’si alınır.
* Dersin veya öğrencinin olmaması durumunda hata mesajı çıkarak ana menüye yönlendirilir.
* İşlemler yapıldıktan sonra öğrenci, ders, kayıt sayfasında gerekli güncellemeler yapılır.



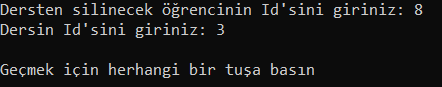




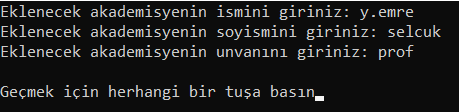




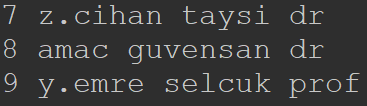
* + 1. **Öğrenciyi Dersten Sil**

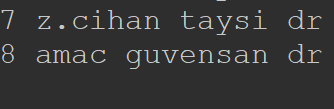


* Öğrencinin ve dersin Id’si kullanıcıdan alınır.
* Öğrencinin, dersin olmaması veya öğrencinin derse kayıtlı olmaması durumunda hata mesajı alınır ve ana menüye dönülür.
  1. **Akademisyen İşlemleri**
     1. **Akademisyen Ekle**



* Yeni eklenecek akademisyenin adı, soyadı, unvanı girilir ve sistem üzerinden kişiye özel Id’si oluşturularak kaydedilir.

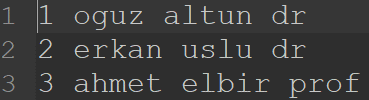


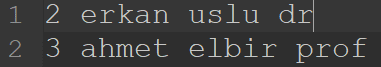


* + 1. **Akademisyen Sil**

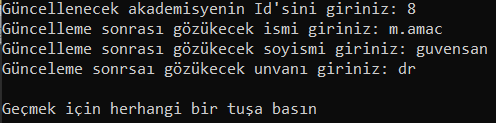


* Silinecek akademisyenin Id’si alınarak gerekli silme işlemleri gerçekleştirilir.
* Verdiği ders varsa o derslerden de id’si silinerek 0’a çevrilir.
* Akademisyenin olmaması durumunda hata mesajı alarak ana menüye döndürülür.

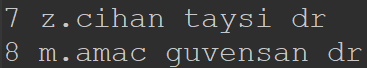




* + 1. **Akademisyen Düzenle**

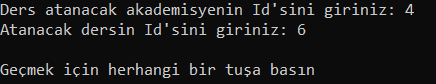


* Gerekli bilgiler girilir ve güncellemeler gerçekleştirilir.
* Girilen Id’ye sahip akademisyenin olmaması durumunda hata mesajı alınarak ana menüye dönülür.

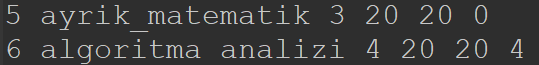


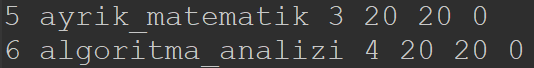


* + 1. **Akademisyene Ders Atama**

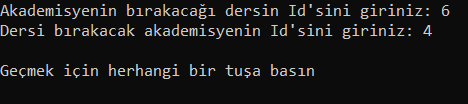


* Dersin ve akademisyenin Id’si alınarak gerekli işlemler yapılır.
* Id’si girilen dersin veya akademisyenin olmaması durumunda hata mesajı verilerek ana menüye dönülür.

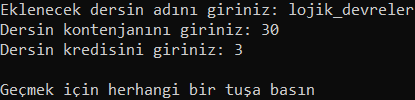




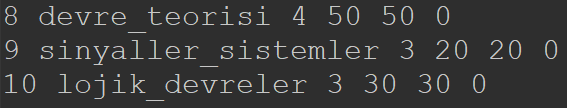
* + 1. **Akademisyene Atalı Dersi Kaldırma**

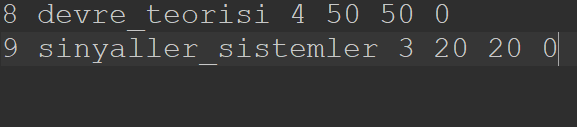


* Dersin ve akademisyenin Id’si alınarak gerekli işlemler yapılır.
* Id’si girilen dersin, akademisyenin olmaması veya akademisyenin dersi vermiyor olması durumunda hata mesajı verilerek ana menüye dönülür
  1. **Ders İşlemleri**
     1. **Ders Ekle**

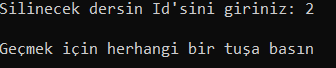


* Eklenecek dersin adı, kontenjanı ve kredisi girilerek ders oluşturulur.



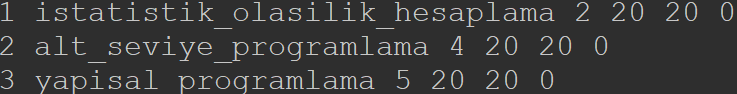


* + 1. **Ders Sil**

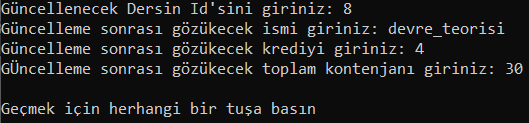


* Dersin Id’si alınarak gerekli işlemler yapılır.
* Girilen Id’nin olmaması durumunda hata mesajı verilerek ana menüye dönülür.

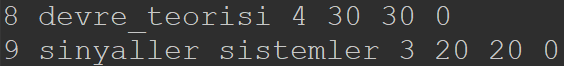


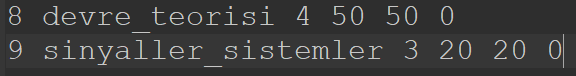


* + 1. **Ders Düzenle**

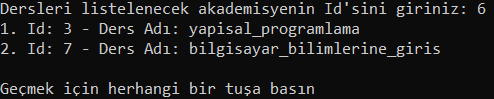


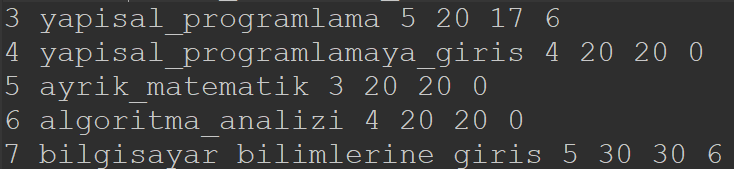
* Gerekli bilgiler alınır ve güncelleme değişen kontenjan ve kredi bilgilerine göre gerekli güncellemeler yapılır.
* Girilen Id’ye sahip bir ders olmaması durumunda hata mesajı verilerek ana menüye dönülür.



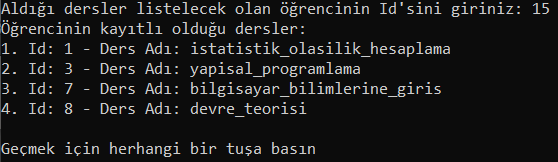


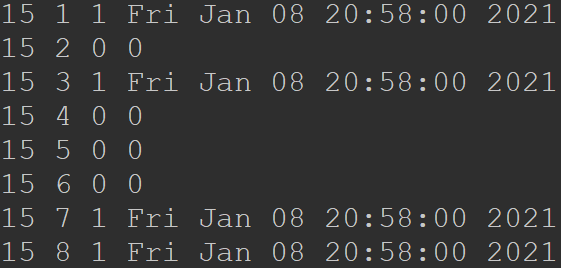
1. **Listeleme İşlemleri Menüsü**
   1. **Akademisyenin Verdiği Dersler**



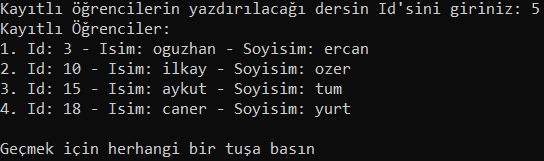


* Id’si girilen akademisyenin verdiği dersler listelenir.
* Akademisyen ders vermiyorsa ders yok uyarısı çıkar ve ana menüye yönlendirir.
* Akademisyen yoksa hata metni çıkar ve ana menüye yönlendirir.
  1. **Öğrencinin Aldığı Dersler**

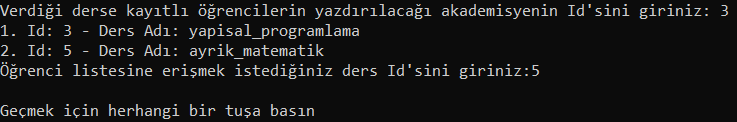




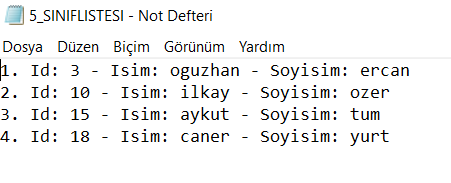
* Öğrencinin Id’si girildikten sonra kaydolduğu dersler listelenir.
* Öğrenci hiçbir derse kayıtlı değilse uyarı mesajı çıkar ve ana menüye yönlendirir.
* Öğrenci yoksa hata metni çıkar ve ana menüye yönlendirir.
  1. **Derse Kayıtlı Öğrenciler**



* Id’si girilen derse kayıtlı öğrenciler ekrana yazdırılır.
* Derse kayıtlı öğrenci yoksa kayıtlı öğrenci yok metni çıkarak ana menüye yönlendirilir.
* Ders yoksa hata metni çıkarak ana menüye yönlendirilir.
  1. **Akademisyenin Verdiği Derse Kayıtlı Öğrenciler**



* Akademisyen Id’si girildikten sonra verdiği dersler listelenir ve tercih sonrasında o dersin DERSKODU\_SINIFLISTESI.txt dosyası oluşturulur ve o dosyaya o dersi alan öğrenciler yazdırılır.
* Akademisyen ders vermiyorsa uyarı metni çıkarak ana menüye yönlendirilir.
* Akademisyen yoksa hata metni çıkar ve ana menüye yönlendirilir.
* Dersi alan öğrenci yoksa dosya içerisine dersi alan öğrenci yok yazdırılır.



1. **Fonksiyonlar** 
   1. **ogrenciDosyaYazdirma**

Öğrenci listesi bulunan dosyaya structta tutulan verilerin “w” modunda yazıldığı fonksiyondur.

* 1. **dersDosyaYazdirma**

Ders listesi bulunan dosyaya structta tutulan verilerin “w” modunda yazıldığı fonksiyondur.

* 1. **akademisyenDosyaYazdirma**

Akademisyen listesi bulunan dosyaya structta tutulan verilerin “w” modunda yazıldığı fonksiyondur.

* 1. **kayitDosyaYazdirma**

Öğrenci – ders kayıt listesi bulunan dosyaya structta tutulan verilerin “w” modunda yazıldığı fonksiyondur.

* 1. **ogrenciIlkKayit**

Öğrencilerin bulunduğu dosyadaki verilerin structta kopyalandığı fonksiyondur.

* 1. **ogrenciEkle**

Öğrenci dosyasına ve structa öğrenci eklenmesini sağlayan fonksiyondur.

* 1. **ogrenciSil**

Öğrenci dosyasından ve structan öğrenci silinmesini sağlayan fonksiyondur.

* 1. **ogrenciGuncelle**

Öğrenci dosyasında ve structta öğrenci verilerinin güncellenmesini sağlayan fonksiyondur.

* 1. **dersIlkKayit**

Ders dosyasında bulunan verilerin structta kopyalandığı fonksiyondur.

* 1. **dersEkle**

Ders dosyasına ve structta ders eklenmesini sağlayan fonksiyondur.

* 1. **dersSil**

Ders dosyasından ve structtan ders silinmesini sağlayan fonksiyondur.

* 1. **dersGuncelle**

Ders dosyasında ve structta bulunan verilerin güncellenmesini sağlayan fonksiyondur.

* 1. **akademisyenIlkKayit**

Akademisyen dosyasında yer alan verilerin structta kopyalanmasını sağlayan fonksiyondur.

* 1. **akademisyenEkle**

Akademisyen dosyasına ve structta yeni akademisyen eklenmesini sağlayan fonksiyondur.

* 1. **akademisyenDersKayit**

Akademisyenin verdiği derslerin eklendiği fonksiyondur.

* 1. **akademisyenDersSil**

Akademisyenin verdiği derslerin kaldırıldığı fonksşyondur.

* 1. **akademisyenSil**

Akademisyen dosyasından ve structtan akademisyen silmeye yarayan fonksiyondur.

* 1. **akademisyenGuncelle**

Akademisyen dosyasında ve structta akademisyen verilerinin güncellenmesini sağlayan fonksiyondur.

* 1. **ogrenciDersKayitDosyaOlusturma**

Programın başında girilen öğrenci ve ders dosyasına göre şemalı bir şekilde öğrencilerin derse kayıt bilgilerinin tutulacağı dosyayı oluşturan fonksiyondur.

* 1. **ogrenciDersKayit**

Öğrencinin derse kaydının yapılmasını sağlayan fonksiyondur.

* 1. **ogrenciDersSil**

Öğrencinin kayıtlı olduğu dersten kaydının silinmesini sağlayan fonksiyondur.

* 1. **akademisyenDerslerListe**

Akademisyenin verdiği derslerin ekrana yazdırıldığı fonksiyondur.

* 1. **dersKayitliOgrenciListe**

Derse kayıtlı öğrencilerin ekrana yazdırıldığı fonksiyondur.

* 1. **ogrenciKayitliDersListe**

Öğrencinin kayıtlı olduğu derslerin ekrana yazdırıldığı fonksiyondur.

* 1. **akademisyenDerseKayitliOgrenciListe**

Akademisyenin verdiği derslerin listelendiği ardından verdiği derslerden birinin seçilerek o derse kayıtlı öğrencilerin o dersin DERSKODU\_SINIFLISTESI.txt dosyasına yazdırıldığı fonksiyondur.

* 1. **dosyaElementSayma**

Dosyalardaki satır sayısını hesaplayan fonksiyondur.

* 1. **tarihHesapla**

Anlık tarihin ve saatin hesaplanmasını sağlayan fonksiyondur.

* 1. **terminalBekletme**

Terminaldeki verilerin okunabilirliğinin artması için kullanılan fonksiyondur.

* 1. **binaryStdSearch**

STD structındaki Id’lerin bulunması için kullanılan binary search fonksiyonudur.

* 1. **binaryDersSearch**

DERS structındaki Id’lerin bulunması için kullanılan binary search fonksiyonudur.

* 1. **binaryAkSearch**

AK structındaki Id’leri bulunması için kullanılan binary search fonksiyonudur.

* 1. **binaryDersSorumluSearch**

DERS strcutındaki akademisyen Id’lerinin bulunması için kullanılan binary search fonksiyonudur.