

# SINIF PLANLAMA SİSTEMİ

Oğuzhan Türker, Ata Güllalan

Bilgisayar Mühendisliği Bölümü  
Kocaeli Üniversitesi

[oguzturker8@gmail.com](mailto:oguzturker8@gmail.com), [xavaneo@gmail.com](mailto:xavaneo@gmail.com)

## Özet

Programımız, üniversite öğrencilerini seçilen atama yöntemine göre istenilen sınıflara ayırarak, sistematik bir sınıf planı oluşturur. Program, eşit dağılım ve en az sınıf dağılımı olmaz üzere iki atama metoduna sahiptir. Metin belgesi içinde bulunan öğrenci bilgilerini bu metotlara göre öğretim türü baz alınarak sınıflara ayırır ve ayrılan sınıflara atanan öğrenci bilgilerini her sınıfa ait oluşturulan metin belgelerinin içlerine yazar. Sınıf planlama programı, projede istenilen tüm işlemleri doğru bir şekilde yapmaktadır.

## 1.Giriş

Sistematik sınıf planlama programı, metin belgesinde bulunan “öğrenci numarası”, “ad”, “soyad”, “kayıt türü”, ve “kayıt sırası” olmak üzere öğrencilere ait olan beş bilgiyi işleyerek sınıflandırır.

Öncelikle metin belgesinde birden fazla aynı ad ve soyada sahip öğrencileri bulup listeden siler. Daha sonra üniversiteye yeni başlayan 2017 girişli öğrencilere kayıt sırasına göre numara atayıp, dersi alttan alan öğrenciler üstte bulunmak üzere tüm öğrencileri numaralarına göre sıralar. Öğrencileri sıralarken, önce birinci öğretim öğrencilerini numaralarına göre sıralar daha sonra ikinci öğretim öğrencilerini kendi arasında numaralarına göre sıralar.

Sıralanan öğrenciler, belirtilen metotlar ile kullanıcının girdiği sınıflara ayrıştırılıp, sıralı bir şekilde ilgili sınıfın metin belgesine yazdırılır. Program üniversitelerin öğrencilerini istenilen ve doğru şekilde sınıflara ayırmasında kolaylık sağlar.

## 2.Temel Bilgiler

Program C programlama dilinde geliştirilmiş olup, tümleşik geliştirme ortamı olarak “CodeBlocks” kullanılmıştır.

## 3.Tasarım

Sınıf planlama sisteminin programlanma aşamaları altta belirtilen başlıklar altında açıklanmıştır.

### 3.1 Algoritma

Program, öncelikle metin belgesinde bulunan öğrencilerin bilgilerini programa çeker. Kendi içinde arama yaparak aynı isimdeki öğrencileri bulup, fazlalıkları siler.

Yine kendi içinde yeni kayıt olan öğrencileri bulup onlara öğretim türü ve kayıt sıralarına göre numara atar. Örneğin ; x öğrencisi 1.öğretim öğrencisi olup kayıt sırası da 12 olsun. Program bu yeni öğrenciye 17 kayıt yılı 01 öğretim türü ve 012 kayıt sırası olmak üzere 1701012 numarasını atamaktadır.

Ardından program kullanıcıya listeye yeni öğrenci ekleyip eklememek istediğini sorar. Kullanıcı listeye yeni öğrenci ekledikten sonra program öğrenci listesini güncelleyip, öğrencileri numara ve öğretim türlerine göre sıralar.

Sıralanan öğrenciler aşağıdaki metotlara göre sınıflara ayrılır.

### 3.1. A Eşit Sınıf Dağılımı;

Kullanıcıdan alınan sınıf kapasiteleri bir dizi içinde saklanır. Tüm sınıfların kapasiteleri toplanır ve sınıf sayısına bölünür. Sınıf kapasitelerinin aritmetik ortalamasından küçük olan sınıflara önce öğrenciler sırasıyla atanır. Atanılan sınıf tüm sınıf kapasiteleri toplamından çıkarılıp, çıkarılan sınıf sayısı kadar da toplam sınıf sayısından düşülür.

Daha sonra kalan sınıf kapasitesi, kalan sınıf sayısına bölünür ve kalan tüm sınıflara bölüm kadar öğrenci atanır. Bölme işleminden eğer kalan olursa kalan sayı kadar öğrenci sırayla kalan sınıflara birer birer atanır. Kapasitesi dolan sınıf atlanır kalanlara doldurulup, işlem tamamlanır.

#### 3.1.B En Az Sınıf Dağılımı;

Kullanıcıdan sınıf sayısı ve kapasitesi alınır. Sınıf kapasiteleri içinde arama yapılır. En büyük olan sınıf bulunur ve atama ilk o sınıftan yapılır. En büyük sınıfa öğrenciler atandıktan sonra, tüm öğrencilerden atanan sınıfın kapasitesi düşülür. Sınıf atandıktan sonra sınıfın değeri saklanan dizi içinde sıfırlanır ve kalan kapasiteler içinde tekrar arama yapılarak, en büyük sınıf bulunur.

Onun içine de öğrenciler atanır ve o sınıfın da boyutu tüm öğrencilerin sayısından çıkarılır. Bu işlem öğrencilerin sayısı bulunan n.en büyük sınıf sayısından küçük olana kadar devam eder, artık öğrenci sayısı bulunan sınıftan küçük olunca, kalan öğrenciler o sınıfa atanıp, varsa kalan küçük sınıflar boş bırakılır.

### 3.2 Kullanılan Fonksiyonlar

*void ogrOku();*

Metin belgesindeki öğrencileri satır satır okur ve ogrParse'a gönderir.

*void ogrParse();*

Öğrencileri boşluğa göre ayırıp ogrOlustur'a gönderir.

*struct ogr \*ogrOlustur;*

Öğrencileri struct dizisine ekler.

*void ogrListele();*

Listele değişkeni 1 ise öğrencileri ekrana yazdırır.

*void ogrSilme();*

Aynı isimdeki öğrencileri listeden siler, isim soyismi silinenEkle'ye gönderir.

*void silinenEkle();*

İsim daha önceden de silinmediyse listeye yeni değer atar, daha önceden de silindiye listedeki değeri bir arttırır.

*void ogrSiral(int i);*

Öğrencileri numara veya kayıt sırasına göre sıralar.

*void ogrNumaralandir();*

Numarası olmayan öğrencilere numara atar.

*void siniflariAl();*

Kullanıcıdan sınıf kapasitelerini alarak bir diziye atar.

*void yazdir();*

Ayrılan öğrencileri kendi sınıflarına yazdırır.

*void esitDagilim();*

Öğrencileri eşit sınıf dağılımı yöntemiyle sınıflandırır.

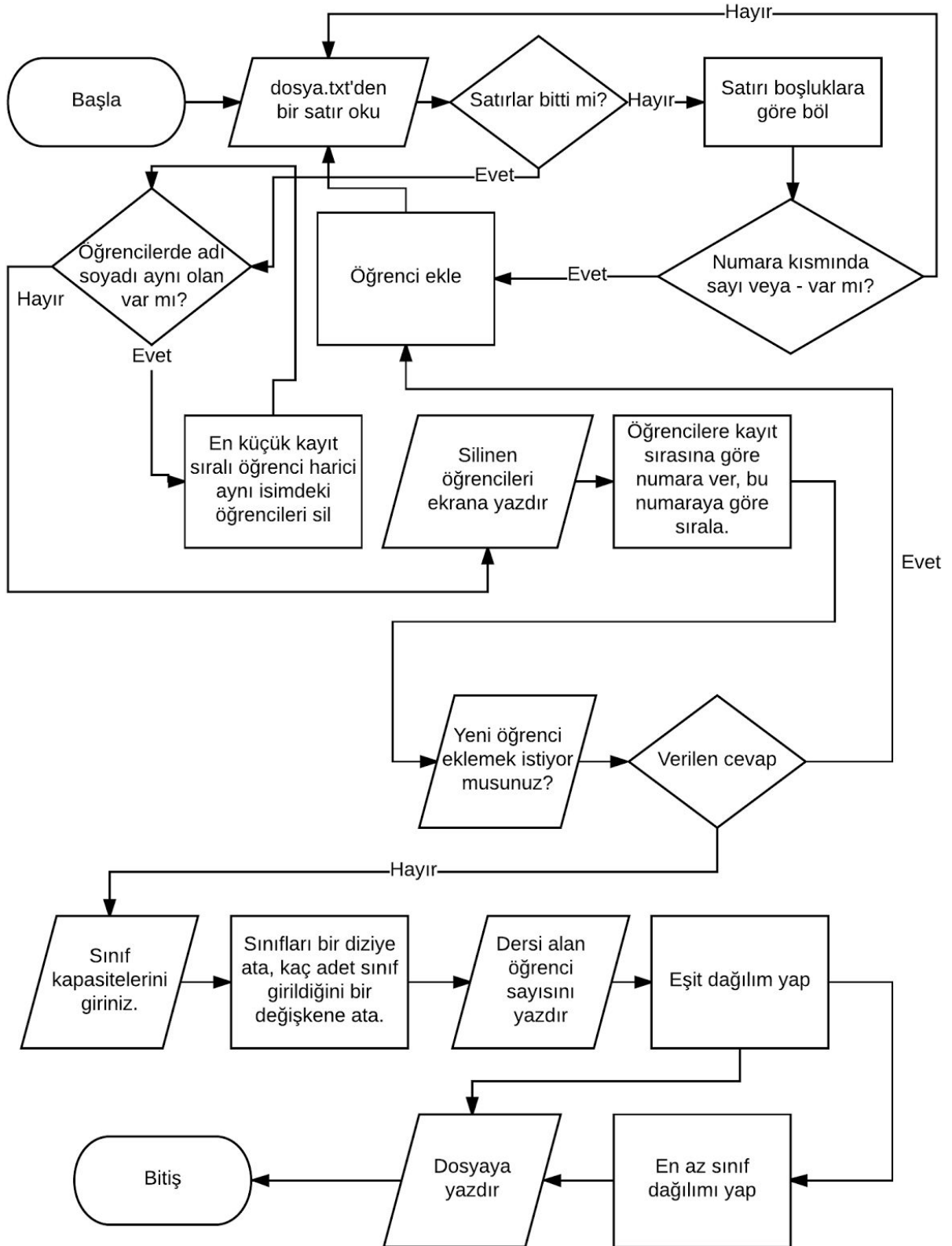
*void enAzDagilim();*

Öğrencileri en az sınıf dağılımı yöntemiyle sınıflandırır.

*void ogrEkle();*

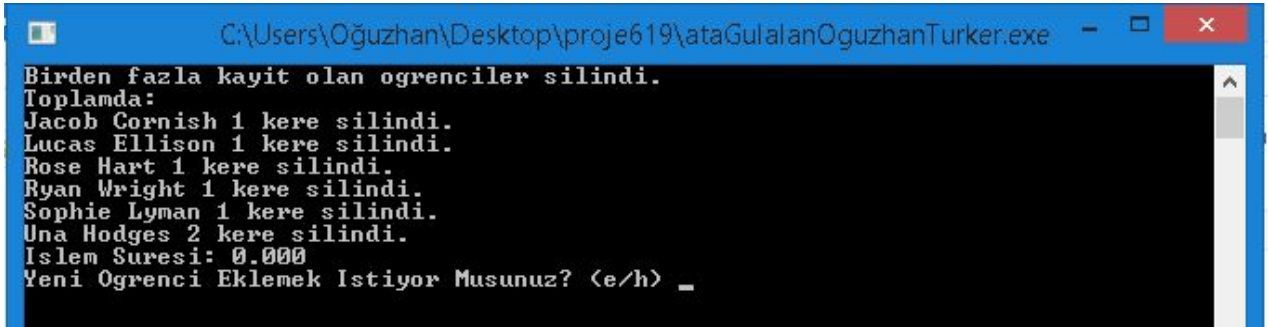
Listeye yeni öğrenci girişi yapar.

### 3.4 Akış Diyagramı

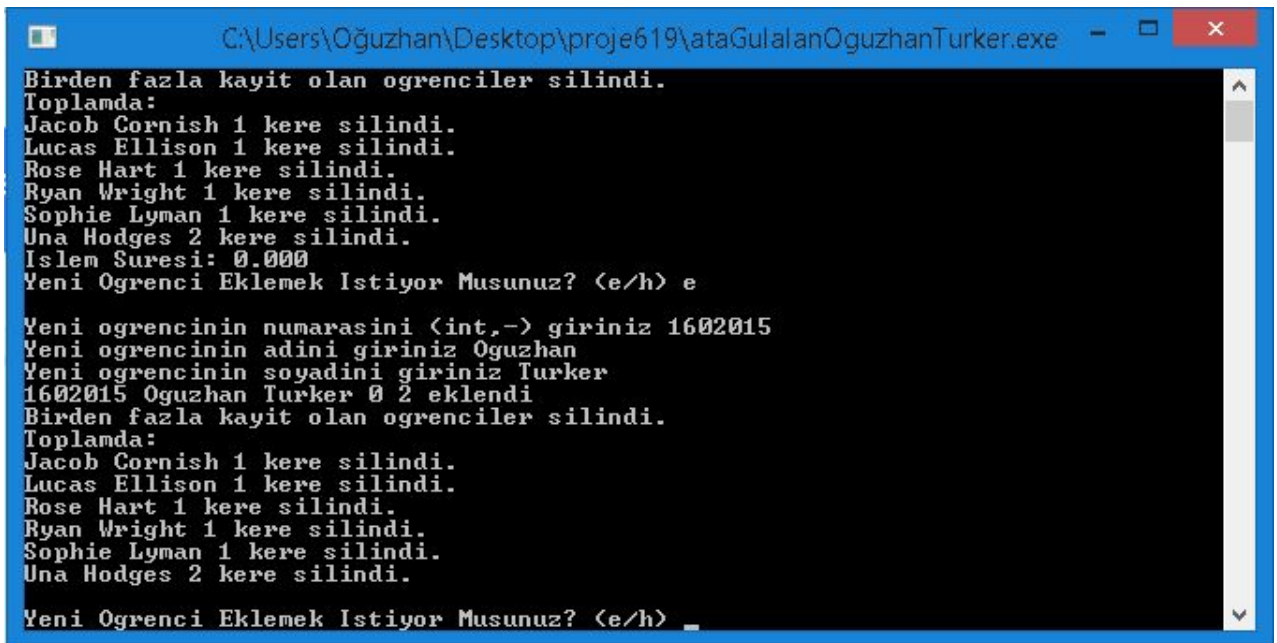


Şekil 1 (Akış Diyagramı)

#### 4. Ekran Çıktıları



Şekil 2 (Aynı İsimli Öğrencileri Silme)



Şekil 3 (Yeni Öğrenci Ekleme)

#### 5. Kaynakça

1. [yazdir() -> snprintf()]  
Convert integer to be used in strcat -  
Stack Overflow  
<https://stackoverflow.com/questions/2805243/convert-integer-to-be-used-in-strcat>  
(Erişim Tarihi: 24.10.2017)