**Sınıf Planlama Sistemi**

SEMA DEMİR - 150201148

[sema.demir.1997@gmail.com](mailto:sema.demir.1997@gmail.com)

**Proje özeti**

Bu proje içinde yapmamız istenen bir Sınıf Planlama Sistemidir. İlk başta elimizde karışık olarak yazılmış bir isim listesi vardır. Bizden istenen bu kişilerin isimlerinin birer kez yazması, numaralandırılması, öğretim türlerine göre de ayrılıp sınıflandırmanın sağlıklı bir şekilde yapılabilmesidir. Bu bağlamda sınıfları hazırlarken öncelikli olarak daha önceden kaydı alınan öğrenci listesini güncellemek gerekiyor ki aynı kişiyi birden fazla yazmayalım.(sil fonksiyonu) Güncellemeyi yaparak tekrarEtmeyenler.txt’inde güncel halini kaydettim. 1. Ve 2. Öğretimleri ayırıp onlara da ayrı ayrı dosya oluşturdum.(bol fonksiyonu)Yeni giriş öğrencilere de numara ataması yaptım.(numAta ve numAtaIki fonksiyonları ile) Bu sınıflara yeni öğrenci kaydı da alınabilmesi için (ogrenciEkle) fonksiyonunu oluşturdum. Son olarak da (sinifOlusturma) fonksiyonunu kullandım. Bu fonksiyon da bize iki seçenek sunuyor. 1. Seçenek ile Eşit Dağılımlı bir sınıf oluşturuyoruz. 2. Seçenek ile de en az sınıfı oluşturuyoruz. Böylece sınıfı güncelledik, öğrencilere numara ataması yaptık, yeni kayıt aldık, yeni öğrenci ekleyebildik ve iki farklı seçenek ile de sınıflarını oluşturduk.

1. **Giriş**

Önerilen projenin konusu bir üniversite için bölümün sınıf düzenini sağlıklı bir şekilde sağlamaktır. Gelen kişilerin bilgilerine göre sınıflanmasını sağlamak, böylece düzenli bir bölümün olması temel hedeftir.

Daha önceden kimlerin böyle projeler yaptığını görmek için Google’da aratma yaptım ve Github’da uygun kodların olup olmadığını inceledim. Bu konu hakkında çalışma yapmış kişilerin olduğunu gördüm. Bu kişiler yoğunluklu olarak yaptıkları uygulamaları daha çok ortaokul sınıflarına uygun olacak şekilde yapmışlar, bunu deneyimlemiş oldum.

Projenin açıklamasında verilen pdf dosyasını dikkatli bir şekilde fazlaca okudum ve adım adım kodlarını yazdım. Adımlar arasında herhangi bir atlama yapmadım ve her kelimesine ayrı ayrı dikkat ettim.Projede herhangi bir eksiklik çıkmaması adına projeyi ve kodu defalarca kez analiz ettim. Ayrıca char ve int arasındaki byte farklarının çalışma süresine olan etkisine de projede fazlaca dikkat etmiş bulunmaktayım. Böylece projenin kendimce eksik olmadığını düşünüyor, herhangi bir aksilik çıkmaması durumunda projeden tam puan almayı hedefliyorum.

**2.Temel Bilgiler**

Projede C dilini kullandım ve derleyici olarak CodeBlocks’u tercih ettim. Yedeklerini tutmak gibi amaçlar için de Notepad’den faydalandım. Proje raporunun daha ayrıntılı olması için bir akış şeması çizmeye karar verdim ve bunun için de ClickCharts isimli bir uygulamadan yararlandım. C dili hakkında bilgi almak için internet üzerinden ufak bir araştırma sonucu;

-Gerçek programcıların kabul ettiği dil

-Farklı bilgisayarlara kolayca uyum sağlaması

-Yapısal ve orta düzey bir dil olması

Gibi temel özelliklerinin olduğunu öğrendim.

“C Programlama dili, orta seviye bir dildir.Makine diline (işlemcinin anladığı 0 ve 1’lerdeden oluşan dil) yakın olan programlama dilleri ile yazılan kaynak kodları uzundur, derlenmiş halleri kısadır ve çalışma hızları çok hızlıdır. Hata düzeltmesi zordur. İşlemcinin daha çok anlayabileceği bir şekilde yazıldığı için en performanslı programlar düşük seviye programlama dilleri ile yazılır.

Yüksek seviye dilleri ile yazılmış kodların kaynak kodları kısadır, derlenmiş halleri uzun olmaktadır. Derleyicinin hata tespiti ve düzenleme özellikleri gelişmiştir. Yüksek seviye bir programlama dili ile yazılan programlar daha yavaş çalışmaktadır.”

Paragrafı da C hakkındaki en temel verileri özetliyor bulunmaktadır.

**3.Diğer Bölümler**

**3.1 Değişkenler**

Projedekullanılan temel değişkenler öğrenci numarası, ismi, soyismi, kayıt sırası ve öğrenci türüdür. Bu değişkenleri kullanarak sınıf oluşturmamız istendiği için geri kalan her şey bu temel değişkenlere entegre edilerek düzenlenir.

**3.2 Fonksiyonlar**

Toplamda 6 adet fonksiyon kullandım. Bu alt başlık içerisinde fonksiyonları teker teker açıklayacağım.

Sil fonksiyonunun amacı tekrar eden kayıtları silmektir. Bu sebeple öncelikle tüm kayıtların olduğu dosyayı okumak gerekiyor. Devamında bir karşılaştırma yapmamız gerekiyor. Burada ad, soyad ve öğretim türünü harf harf karşılaştırıyoruz. 1 harf bile farklıysa bunlar farklı kişilerdir. Fakat hepsi doğruysa aynı kişinin birden fazla kaydı var demektir. Birden fazla kaydı olan kişileri ekrana basarız. Aynı olmayanları da tekrarEtmeyenler.txt’ine yazarız. İlk dosyayı okumak için r (read) komutunu kullarınırız. tekrarEtmeyenler dosyasını yeni oluşturup içine bir şeyler yazacağımız için w(write) komutunu kullanırız.

Bol fonksiyonunun amacı tekrarsız öğrenci kayıtlarını 1 ve 2. Öğretim olarak ayırmaktır. Öğrenci kayıtlarının olduğu dosyada bize öğretim türleri roma rakamlarıyla (I ve II) verilmiştir. Yeni giriş öğrencilerin de numarası olmadığı için – ile gösterilmiştir. Integer olarak işlem yapmak daha kolayıma geldiği için I ve II ‘ yi 1 ve 2 olarak, - yi de 0 değiştirdim. Öğretim türü 1 olanları birinciOgretim.txt, 2 olanları ikinciOgretim.txt dosyalarına w komutuyla yazdırdım. Böylece tekrarsız kayıtları öğretim türlerine göre bölmüş oldum.

numAta fonksiyonunun amacı birinci öğretim yeni giriş öğrencilerine numara atamaktır. Bol fonksiyonunda herkese 0 yazmıştık. Burada öğrencilerin kayıt sırasına göre numara verdim. Numara atama sistemi şu şekildedir;

2017 girişli olduğu için numara 17 ile başlayacaktır. 1. Öğretimler 01 ile 2. Öğretimler 02 ile gösterilecektir. Devamında da kayıt sırası devreye girecektir. Yani 1. Sıradaki öğrencinin numarası 1701001 olacaktır. Bu kayıtları da numBirinci.txt adlı dosyaya w komutuyla yazdırdım.

numAtaIkı fonksiyonunun amacı ikinci öğretim yeni giriş öğrencilerine numara atamaktır. Numara atama sistemini bir önceki fonksiyonda açıklamıştım. Aynı şekilde olup sadece öğretim kısmı için 01 yerine 02 kullanılacaktır.

OgrenciEkle fonksiyonun amacı numaralandırılmış şekilde ayırdığımız dosyalara yeni öğrenci girişi yapmaktır. Burada işlem yaparken sıralamamız şu şekildedir;

İlk olarak öğretim türünü belirleriz. 1. Öğretim ise eklenecek olan öğrencinin numarasını ismini soyismini isteriz, a komutu ile numBirinci.txt soyasına yazarız. W komutu içindekileri silerek yazar, a komutu eskiyi tutup üzerine yazar bu sebeple a yı kullanırız.

Aynı işlemi 2. Öğretim için de yaparız ve numIkinci.txt dosyasına kaydederiz.

SinifOlusturma fonksiyonun amacı istenilen verilere göre sınıf oluşturmaktır. Burada ilk istenilen kaç tane sınıf olduğudur. Devamında bu sınıfların kapasitesi istenir. Kapasite öğrenci sayısından küçük verilirse program hata verir. Kapasite verildikten sonra bizden tercih yapmamız istenir. Bu tercihler eşit dağılımlı mı yoksa en az sınıf mı olduğudur. Eğer tercihimizi eşit dağılımdan yana kullanırsak;

Öncelikle sınıf kapasitelerini küçükten büyüğe sıralarız. Daha sonra ortalamaya bakarız. Eğer öğrenci sayısının sınıflara göre ortalaması ilk sınıfın kapasitesinden küçükse sınıfı doldururuz. Sınıf sayısını bir azaltıp, kalan öğrencilerle ortalamasına bakarız. Küçükse aynı işlemlere devam ederiz. Eğer kapasite büyükse tüm sınıflara ortalamayı yerleştiririz.

Eğer tercihimizi en az sınıftan yana kullanırsak;

Öncelikle sınıf kapasitesini büyükten küçüğe doğru sıralarız. En büyük kapasiteli sınıflardan başlayarak sınıfları kapasiteleri kadar doldururuz. Eğer artan sınıf olursa program bize doldurulan sınıfların kapasitesini yazdıktan sonra bundan sonraki sınıflarda öğrenci bulunmamaktadır çıktısı verecektir. Böylece sınıfları da oluşturmuş oluruz.

Kaç tane sınıf olacağını bilmediğimiz için sınıfları dinamik bir şekilde oluşturmamız gerekmektedir. Yani kaç sınıf var ve hangi sınıfta kaç kişi var, kimler var bunları görmek için dinamik bir yapımız mevcut.

Örneğin;

5 tane sınıf oluşturmak istediysek 1. Sınıf 1. Öğretim, 1. Sınıf 2. Öğretim … şeklinde toplam 10 tane sınıf oluşur.

Main fonksiyonunda da bu işlemler sırasıyla

sil();

bol();

numAta ();

numAtaIkı();

ogrenciEkle();

sinifOlusturma();

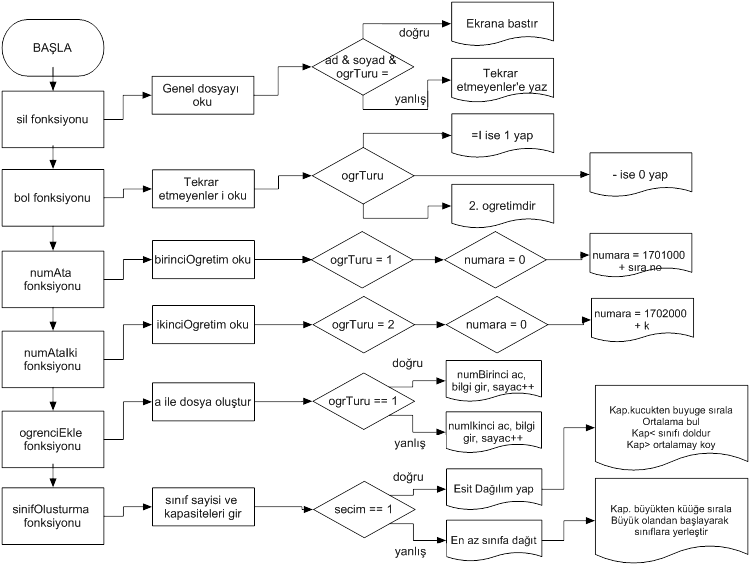
olarak başarılı bir şekilde derlenir.

**4.Sonuçlar**

İlk olarak yaşadığım problem grup arkadaşımla alakalıydı. Tekrardan yazılması gerektiğini bilmiyorduk bu sebeple tarihi kaçırdık ve projeyi tek başıma yapmak durumunda kaldım.

Projede en çok yoran kısım char ve integer arasındaki dönüşümler sebebiyle derleyicinin vermediği hatalar oldu. Ekrana herhangi bir çıktı verilmiyor üstüne hata da vermiyordu. Bu benim için gerçek anlamda sıkıntıydı çünkü nereyi düzelteceğimi bilmiyordum ve ekrandan çıktı alamıyordum. Kodu saatlerce inceledikten sonra hatanın char ve integer arasındaki dönüşümden kaynaklandığını anladım ve orayı düzeltince istediğim çıktıları almaya başladım.

Diğer bir kısım ise dinamik yapıları bilmiyor oluşumdu. Dinamik yapı oluşturma kısmını uzun uzun araştırdım fakat hala anlayamadım. Bu konu hakkında bilgili hocalarımıza danıştım, İngilizce kaynaklar inceledim. Yoğun süren uzun çalışma saatleri sonucu dinamik yapıyı oluşturmayı başardım. Herhangi bir eksiğimin olmaması için proje açıklamasının verildiği dokümanı uzun uzun inceledim ve kaçırdığım yer olmaması için adım adım takip ettim. Yoğun süren çalışmalarım sonucu projeyi tamamladım. Herhangi bir aksilik çıkmazsa tam puan almak hedefimdir.

****

**5.Kaynakça**

[1] <http://80.251.40.59/science.ankara.edu.tr/ozbek/cson.HTM>

[2] <https://ckaynak.com/c-programlama-diline-giris-848#ozellikleri>

[3] Yorulmaz,M ve Yorulmaz,S “Programlamayı C ile Öğreniyorum” Palme Yayınları 366-380

[4] Şeker,ŞE “C ile Dosya İşlemleri” <http://bilgisayarkavramlari.sadievrenseker.com>