

İK YÖNETİM SİSTEMİ

Aziz GÜLTEKİN, Enes GELMEZ

Bilişim Sistemleri Mühendisliği Bölümü
Kocaeli Üniversitesi

181307047@kocaeli.edu.tr ve 171307066@kocaeli.edu.tr

Özet

Programımızda temel dosya yapılarını kullanıp, istenilen verilerin yazılması ve çekilmesi işlemlerinin her kullanıcının TC numarasını kullanarak .txt uzantılı bir dosyada o kişinin verilerini tutarak ilgili işlemlerin gerçekleşmesini sağlamaktadır. Böylelikle programımız çalıştığı zaman TC numaralarına göre ağacı oluşturup listeleme kısmında preorder, inorder, postorder gibi sıralama unsurlarını gerçekleştirmektedir. Önemli diğer konulardan biri ise işçi ararken kullanılacak olan filtreleme yöntemidir. Bu yöntemle tüm kişiler arasından belirli özelliklere sahip olan kişilerin kolayca listelenmesi sağlanabilmektedir.

1. Giriş

Bu program bir çok yönüyle iş arayanların veya işyerinde çalışacak olan kişileri bulma işini kolaylaştırmak amacı ile yazılmıştır. İş arayacak olan kişi gerekli tüm bilgilerini girdikten sonra o kişi adına belirli bir uzantıda bilgilerini .txt uzantılı bir dosyada tutup, işçi arayan bir firmanın herhangi bir filtreleme işleminde bu işçi gereksinimlerini karşılıyorsa kolay bir şekilde irtibata geçip gerekli işlemler daha seri bir şekilde ilerletilebilir.

2. Temel Bilgiler

Microsoft Visual Studio Community 2019 version 16.7.6

.Net Framework 4.8.03752

Windows Form Application

Kullanılan kütüphaneler:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.IO;
```

System.IO kütüphanesi c# ile dosyalama yapabilmemiz için kullandığımız kütüphane. Bu kütüphaneye ait kodların açıklamaları:

File.Delete(dosya_yolu); Parantez içerisine girilecek yoldaki dosyanın silinmesini sağlar.

File.Exists(dosya_yolu); Parantez içerisine girilecek yoldaki dosyanın varlığını sorgular ve geriye true ya da false olarak değer döndürür.

```
bool kontrol = File.Exists(dosya_yolu);
if (kontrol == true)
{
}
```

Böyle bir kod bloğu ile dosya yolundaki dosya var ise if bloğu içerisinde istenilen bir eylem gerçekleştirilebilir.

```
StreamReader sr = File.OpenText(dosya_yolu);
sr.ReadLine();
```

Belirli yoldaki dosyanın içerisinden herhangi bir veriyi okumanızı sağlar.

```
StreamWriter sw = new StreamWriter(dosya_yolu);
sw.WriteLine();
```

Belirli yoldaki dosyanın içerisine veri girişi yapmanızı sağlar.

3. Geliştirilen Mimari

3.1 Kullanılan sınıfların ve metotlarının kısa açıklamaları;

Bu bölümde programımızın büyük bir kısmını oluşturan metotların ve sınıfların açıklamalarını yapacağız. İlk olarak sisteme kayıt olacak kişilerin (başvuru.cs) form ekranındaki sınıf ve metotlardan başlayalım.

DosyaKaydet(): Bu metot ile textboxlara girilen verilerin ilgili yol üzerinden kişinin TC numarası ile .txt dosyası oluşturup içerisini girilen bilgilerin düzenli bir şekilde aktarılmasını sağlıyor.

VerileriGetir(): Bu metot sayesinde kişi kendi bilgilerini ilgili textboxlara aktarabilir ve böylelikle bir değişiklik yapacağı zaman sadece o kısımda değişiklik yapılabilir.

Guncelle(): Kullanıcının herhangi bir bilgisini değiştirmek istediğinde bu metot ile VerileriGetir() metodunu kullandıktan sonra istediği kısımda değişiklik yapıp bu metot ile güncelleme yapılabilir.

VerileriSil(): Artık sistemde verilerinin bulunmasını istemeyen birisi bu metot ile klasörümüzün içinden kendi verilerini silebilmektedir.

Class Dugum: Bu class ile linkedlist kullanmak için girilecek olan verilerin değişkenleri tutulmaktadır. Örneğin “Public long tc;”

Class LinkedList: Girilecek olan verilerin Dugum classının içerisindeki değişkenlere atanmasını sağlar.

Şimdi işçileri arayabileceğimiz(ik.cs) form ekranındaki sınıf ve etotlardan devam edelim.

TumverileriCek(): Metodun amacı sistemde kayıtlı olan tüm kişilerin bilgilerini ilgili listbox’ın içine doldurmaktır.

IstenilenveriyiCek(): Dosyanın yolunu belirledikten sonra o TC numaralı .txt dosyasının içindeki verileri çekmemizi sağlıyor.

Listele(): En önemli kısımlardan birisi de burada. İstenilen verilerin birbirleri ile bağlantılı olarak listeleme aşamasıdır. Örneğin işçi arayan birisi İngilizce bilen, yaşı minimum 20 olan ve minimum 2 yıl tecrübesi olan birisini aradığı zaman ilgili yerlere bu verileri girip listele komutunun kullanılması durumunda belirli if bloklarıyla oluşturduğumuz filtreleme aşamasını geçtikten sonra IstenilenveriyiCek() metodu ile o kişileri listelemektedir.

txtTemizle(): Gerekli textboxları temizleme işlemini yapar.

Eskihalinedondur(): Kaydet butonundaki işlemler yapıldıktan sonra gerekli textboxların enable özelliklerini değiştirir ve form ekranının ilk haline dönmesini sağlar.

Deneyim_sure_bul(): Burada listeleme aşamasında kişinin belirli yıl deneyim süresi olması durumunda verileri çekebilmemiz için o kişinin ne kadar deneyim süresi olduğunu hesaplatıyoruz.

Yas_bul(): Burada da Deneyim_sure_bul() metodunda olduğu gibi bu metod ile de yaş bulduruyor.

Sec(): Dosyalama yapısı oluştururken preorder, inorder, postorder dizilimlerini farklı .txt lerde tuttuğumuz için form ekranında hangisi seçilirse bu o seçilen dosyanın yolunu gönderen metod

Cikti_Al(): İşçi arayan birisi listeye verilerini çektikten sonra bunların bir çıktısını almak için bu metodu kullanacak. Böylelikle masaüstünde o kişilere ait bir .txt dosyası oluşacak.

Ad_listele(): Sadece adların listelenmesi için oluşturulan bir metod.

Class Agac: Bu sınıf ağacımızın oluşumunda kullanılıyor. Gelen veri (TC) bu alanda köklerine ve dallarına ayrılıp ağaç yapısını oluşturuyor. Bu sınıfın içinde ayrıca preorder, inorder, postorder listeleme işlemleri de bulunmakta.

Class Dugum: Düğümüne ait verilerin barındığı kısım. İçerisinde 3 adet değişken bulunuyor. Bunlar

```
public long veri;  
public Dugum Sol;  
public Dugum Sag;
```

3.2 Kullanılan veri yapılarının küçük örneklerle anlatımı ve kısa açıklamaları [3] [4]

İlk olarak LinkedList kullandık. Bu yapı verilerin kaydedilmesi kısmında kullanılıyor.

Bir sonraki ise Ağaç yapısı. Bu yapı LinkedListe göre biraz daha karışık. İlk olarak ağaç kendisinde kök var mı diye soruyor eğer yoksa çekilen ilk veriyi kök olarak kabul edip daha sonra gelen verileri kurala göre listeliyor. Eğer gelen TC no kökteki veriden daha büyük veya eşit ise sağa doğru ilerliyor eğer daha küçük ise soldan devam ediyor ve böylelikle ağaç oluşuyor. Bunun amacı ise kişilerin fazla olması durumunda veriye daha kolay ve hızlı ulaşmak için ağaç yapısı kullanmak yararlı oluyor.

Bu yapıyı kullanırken Ağaç sınıfının içinde belirli yapılması gereken aşamalar var. Bunların birkaçını açıklayalım. İlk olarak public void Ekle() diye bir metod kullanıyoruz. Bu ağacın her düğümünü oluşturmak için kullanılıyor. Daha öncede bahsettiğim gibi gelen veri ilk olarak buraya gelip burada ağacın ne tarafa doğru şekillenip oluşturulacağına karar veriliyor. Daha sonra preorder, inorder ve postorder metodlarımız ile ağacı nasıl sıralamak istediğimizi belirliyoruz. Örneğin tasarladığımız ara yüz kısmında verilerin listelenmesini preorder metodu ile yapmak istiyorsanız gerekli radiobutonunu seçip ona göre sıralayabilirsiniz. Preorder da sıralama kök-sol-sağ, inorder da sol-kök-sağ, postorder da ise sol-sağ-kök olarak ağacı dolaşmaktadırlar. ReturnKok() metodu ise postorder veya diğer sıralama biçimlerinin içlerine ağacın kökünü göndermemiz için gerekli metod. Ağacın kökünü gönderiyoruz ve kendi içinden kökten başlayarak ilkili adımlara devam ediyor.

```
class Agac  
{  
  
    public Dugum kok = new Dugum();  
    public Agac()  
    {  
        kok = null;  
    }  
    public Dugum ReturnKok()  
    {  
        return kok;  
    }  
    public void Ekle(long v)  
    {  
        Dugum YeniDugum = new Dugum();  
        YeniDugum.veri = v;  
        ik i = new ik();  
        if (kok == null)  
        {  
            kok = YeniDugum;  
        }  
        else  
        {  
            Dugum ebeveyn;  
            Dugum mevcut = kok;  
  
            while (true)  
            {  
                ebeveyn = mevcut;  
                if (v < mevcut.veri)  
                {  
                    mevcut = mevcut.Sol;  
                    if (mevcut == null)  
                    {  
                        ebeveyn.Sol =  
YeniDugum;
```

```

this.derinlik_sayisi = derinlik_s;
                        derinlik_s++;
                        i.label18.Text =
derinlik_s.ToString();
                        return;
                    }
                }
            else
            {
                mevcut = mevcut.Sag;
                if (mevcut == null)
                {
                    ebeveyn.Sag =
YeniDugum;
this.derinlik_sayisi = derinlik_s;
                        derinlik_s++;
                        i.label18.Text =
derinlik_s.ToString();
                        return;
                    }
                }
            }
        }
    }
}

```

3.3 Karşılaşılan problemler ve çözüm yaklaşımları; [1] [2]

Dosyalama yaparken dosyaların hepsini bir arada tutup , istenildiğinde dosyadan bilgi çekmek zor olacağını düşündüğümüz ve ağaç yapısında sayılarla daha kolay bir biçimde dolaşma yapıldığı için dosyalamada herkesin T. C. kimlik numarasına göre ayrı ayrı dosyası olacak şekilde , istenilen bir bilgiye daha kolay ulaşmayı amaçladık.

Kullanıcılar ve admin adında iki adet dosya oluşturduk. Bunun sebebi adminde preorder, inorder ve postorderları T. C. kimlik numarasına göre sıraladığımızda kullanıcılardaki dosyalarda tek bir dosyaya üçünü attığımız da çekmek zor ve karmaşık olabileceğini deneyimlediğimiz için ayrı bir dosyada dolaşma işlemlerini yaptık ve hangisini işaretlerse ona göre çekmeyi sağladık.

Birden fazla atama gerektiren(Örneğin okul adı , iş yeri adı vb.) değerler için form kısmında birden fazla textbox koyduğumuzda hem kötü gözükebileceğini hemde kod kısmında sıkıntı yaratabileceğini deneyimlediğimiz için checkbox yardımı ile doldurmak istediğimiz alanları açabilmeyi sağlayarak bu sorunu ortadan kaldırdık.

Yabancı dil ve ehliyetle yine dört checkboxlı yaptığımızda dosya okutmada ve filtrelemede sıkıntı yarattığı için bir textbox yapmada karar kıldık. Tek textboxa virgül ile ayırarak değer girdiğimizde sorun olmadığını deneyimledik , aratma, ve filtreleme özelliğinde çalışır olduğunu gözlemledik.

Filtreleme kısmında bütün filtrelemeleri tek tek yaptırdığımızda mesela sadece İngilizce bilenleri filtrelediğini gözlemledik bizde bunu daha geniş kapsamlı yapmak için yine aynı örnek üzerinden mesela İngilizce ve Almanca bilen , sadece Almanca bilen , vs. gibi özellik katarak daha geniş ve form tasarımı daha iyi görünür olan bir filtreleme kısmı yapmış olduk.

3.4 Yazılım Geliştirme İçin Harcanan Süreler (kişi ve saat bazında);

Enes: Haftanın 4 günü ve minimum günde 5 saat olmak üzere toplamda yaklaşık 90 saat bu program için çalışma yaptı.

Aziz: Haftanın 4 günü ve minimum günde 4 saat olmak üzere yaklaşık 70 saat bu program için çalışma yaptı.

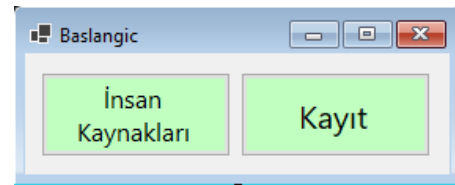
3.5 Dosyalama sisteminin, örnek veriler üzerinden anlatımı(Resim 1.7, 1.8)

Bizim dosyalama sistemimizde proje adında bir ana klasörümüz var ve onun alt dosyaları olarak kullanıcılar ve admin klasörleri mevcut. Bu klasörleme mantığı şöyle oluşmakta, admin klasörünün içerisinde program her çalıştığında kullanıcılar klasöründe bulunan kayıtlı kişilerin listesini preorder, inorder, postorder sıralama mantıkları ile kendi aralarında sıralayıp her biri için yeni bir .txt dosyası oluşturuyor, böylelikle programın ilerleyen aşamasında sıralama mantığı değiştiği zaman uzantımız da değişip istenilen sıralamam mantığıyla sıralama yapıyor. Kullanıcılar klasöründe ise girilen veriler TC no adıyla .txt uzantılı dosyalar oluşturup o kişinin tüm bilgilerini o .txt dosyasında tutuluyor.

Örnek verecek olursak, bir kişi kayıt olmak için gerekli alanlara verilerini girdiğini düşünelim. Daha sonra kayıt butonuna tıkladığında TC no ile kullanıcılar klasörü altında yeni bir txt dosyası oluşturuyor ve içine girilen bilgileri LinkedList mantığı ile kayıt ediyor, eğer daha önceden kayıt olmuş ise yeniden kayıt olamıyor.

4. Kullanıcı Kataloğu

Geliştirilen Programın İşletimi ve Ekran Görüntüleri (en az 1 tane) ile anlatımı



(Resim 1.0)

TC No:

Ad:

Soyadı:

Adresi:

E-Mail:

Tel No:

Doğum Yılı:

Yabancı Dil:

Ehliyet:

Eğitim Bilgileri

Okul Adı	Türü	Bölümü	Başlangıç Tarihi	Bitiş Tarihi	Not Ortalaması
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

İş Deneyimi

Çalıştığı Yer Adı	İş Adresi	İş Tecrübesi Pozisyon veya Görevi	İş Tecrübesi Süresi
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Kayıt Ol

Kayıt Sil

Kayıt Güncelle

Bilgileri Getir

Kapat

Geri Dön

(Resim 1.1)

Kayıt Ol

Kayıt Sil

Kayıt Güncelle

Bilgileri Getir

Kapat

Geri Dön

(Resim 1.2)

İnsan Kaynakları

Ad ile Listeleme

Mezun Durumuna Göre Listeleme

Yabancı Dil Bilenlere Göre Listeleme

Minimum Deneyim Süresine Göre Listeleme

Belirli Yaş Değeri Altındaki Kişiler Göre Listeleme

Belirli Tip Ehliyeti Olan Kişilere Göre Listeleme

☒ InOrder ☐ PreOrder ☐ PostOrder

Eleman Sayısı:

Derinlik:

Arama Yap

Tüm Kişileri Listele

Ad ile Listeleme

Çıktı Al

Geri Dön

(Resim 1.3)

İnsan Kaynakları

Ad ile Listeleme

Mezun Durumuna Göre Listeleme

Yabancı Dil Bilenlere Göre Listeleme

Minimum Deneyim Süresine Göre Listeleme

Belirli Yaş Değeri Altındaki Kişiler Göre Listeleme

Belirli Tip Ehliyeti Olan Kişilere Göre Listeleme

☒ InOrder ☐ PreOrder ☐ PostOrder

Eleman Sayısı: 1

Derinlik:

Arama Yap

Tüm Kişileri Listele

Ad ile Listeleme

Çıktı Al

Geri Dön

TC No: 12345678913
 Ad: Neriman
 Soyadı: Koç
 Adresi: Ankara
 E-Mail: nkoca@gmail.com
 Tel No: 5331234567
 Doğum Tarihi: 1988
 Yabancı dil: Almanca
 Ehliyet: E

Eğitim Bilgileri

Okul adı: Ankara Anadolu Öğretmen Lisesi
 Türü: lise
 Bölümü: Sayısal
 Başlangıç tarihi: 2002
 Bitiş tarihi: 2005
 Not ortalaması: 90

Okul adı: Ege Üniversitesi
 Türü: lisans
 Bölümü: Elektronik Haberleşme Müh
 Başlangıç tarihi: 2005
 Bitiş tarihi: 2010
 Not ortalaması: 70

Okul adı: Ege Üniversitesi
 Türü: yüksek lisans

(Resim1.4)

Basvuru

TC No: 12345678913 Bilgileri Getir

Ad: Neriman

Soyad: Kaç

Adres: Anlara

E-Mail: nkoca@gmail.com

Tel No: 5331234567

Doğum Yılı: 1988

Yabancı Dil: Almanca

Ehliyet: E

Eğitim Bilgileri

Okul Adı	Türü	Bölümü	Başlangıç Tarihi	Bitiş Tarihi	Not Ortalaması
Ankara Anadolu Öğretmen Lis	lise	Sayısal	2002	2005	90
<input checked="" type="checkbox"/> Ege Üniversitesi	lisans	Elektronik Haberleşme Müh	2005	2010	70
<input checked="" type="checkbox"/> Ege Üniversitesi	yüksek lisans	Elektronik Haberleşme Müh	2011	Devam Ediyor	e

İş Deneyimi

Çalıştığı Yer Adı	İş Adresi	İş Tecrübesi Pozisyon veya Görevi	İş Tecrübesi Süresi
Ege Üni Elektronik Haberleşme	İzmir	Araştırma Görevlisi	3
			0
			0
			0
			0
			0

Kayıt Ol Kayıt Sil Kayıt Güncelle Geri Dön Kapat

(Resim1.5)

İnsan Kaynakları

Ad ile Listeleme

Mezun Durumuna Göre Listeleme

Yabancı Dil Bilenlere Göre Listeleme

Minimum Deneyim Süresine Göre Listeleme

Belirli Yaş Değeri Altındaki Kişiler Göre Listeleme

Belirli Tıp Ehliyeti Olan Kişilere Göre Listeleme

☒ InOrder ☐ PreOrder ☐ PostOrder

Eleman Sayısı: 1

Derinlik:

Arama Yap Tüm Kişileri Liste Ad ile Listeleme Çıktı Al Geri Dön

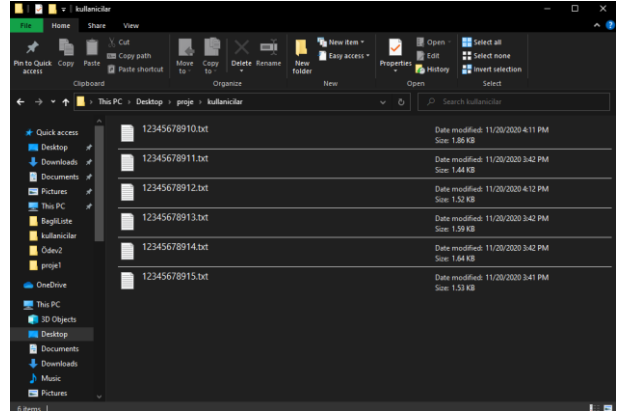
TC No: 12345678910
Ad: Ali
Soyad: Kaya
Adresi: İstanbul
E-Mail: akaya@gmail.com
Tel No: 5321234567
Doğum Tarihi: 1985
Yabancı dil: ingilizce,fransızca
Ehliyet: A2,B

Eğitim Bilgileri
Okul adı: Galatasaray Lisesi
Türü: lise
Bölümü: Sayısal
Başlangıç tarihi: 1999
Bitiş tarihi: 2003
Not ortalaması: 82

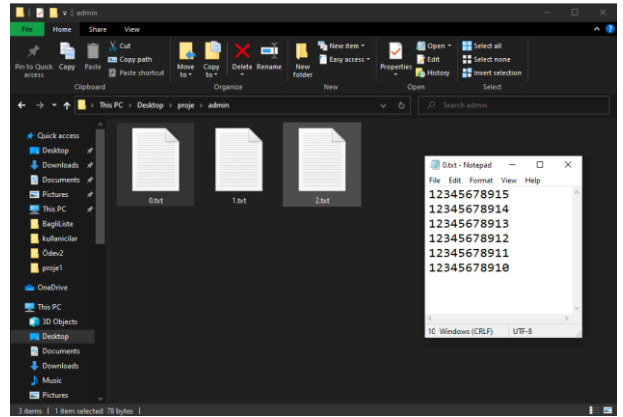
Okul adı: Yıldız Teknik Üniversitesi
Türü: lisans
Bölümü: Bilgisayar Müh
Başlangıç tarihi: 2004
Bitiş tarihi: 2009
Not ortalaması: 80

Okul adı: Yıldız Teknik Üniversitesi
Türü: yüksek lisans

(Resim1.6)



(Resim 1.7)



(Resim 1.8)

Resim 1.0 da bizim başlangıç ekranımız bulunmakta. Burada kayıt butonuna tıklarsak sisteme kayıt olmak için tasarlanan ekran ile karşılaşacağız (Resim 1.1), insan kaynakları kısmı ise kayıtlı olan kişilerin istenildiği şekilde listelenmesini sağlıyor.(Resim 1.2)

Resim 1.2 de görülen butonlarda;

Kayıt Ol butonu girilen bilgilerin ilgili dosya yolunda TC no adıyla bir dosya oluşturup verileri onun içine kaydeder.(Resim 1.5)

Kayıt Sil butonu var olan bir kullanıcının bilgilerini silmesini sağlar. Bu buton direkt dosyayı siliyor.

Kayıt Güncelle butonu kayıtlı olan bir kişinin herhangi bir verisini değiştirmek istemesi durumunda ilk önce “Bilgileri Getir” butonuna tıklayıp daha sonra ilgili textboxların içeriği kayıtlı olan bilgiler ile dolduktan sonra istenilen veriyi değiştirip bu butona bastığında veriler güncellenmiş oluyor.

Kapat butonu uygulamanın kapanmasını sağlar.

Geri butonu ise başlangıç form ekranına (Resim 1.0) dönmemizi sağlar.

İnsan kaynakları ekranındaki butonlardan bahsetmek gerekirse;

Tüm Kişileri Listele butonu, tıklandığı takdirde kayıtlı olan bütün kişilerin listbox'a getirilmesini sağlıyor.

Arama butonu, belirli bir veri aranmak istediğinde ilgili yerleri doldurduktan sonra butona tıklayıp o verilere ait kullanıcıları listelemeyi sağlıyor.(Resim 1.4,1.6)

Ad ile listeleme butonu, kayıtlı olan tüm kişilerin adlarını listelememizi sağlıyor. Eğer inorder seçili ise inorder sıralaması yöntemiyle adları getiriyor.

Çıktı Al butonu ise listboxta olan verilerin hepsini masaüstünde yeni bir txt dosyasının içerisine kayıt ediyor.

5. Sonuçlar

Bu çalışmada hedefimizin hemen hemen hepsini tamamlamış bulunmaktayız. Bazı eksiklikler programın son aşamalarına doğru kendini belli etmeye başladı. Ne kadar düzelmeye çalışsak da bazı eksiklikler programda bulunmaktadır. Örneğin ağacımızın derinliğini hesaplariken hiçbir sorun yaşamadık fakat istenilen elemanın hangi derinlikte olduğunu bulduramadık.

Bir diğer sorunu dosyaları preorder, inorder, postorder ile çekerken yaşadık. Normalde eklediğimiz her eleman dosya kısmında tarihe göre tutuluyor ve en son eklenen kişi sonuncu olarak geliyor. Fakat kullandığımız kod satırında dosyayı kendi belirlendiği default değerlere göre çektiği için veriler hep ya ağacın soluna ya da sağına gidiyor. Örneğin resim 1.8 de postorder olarak sıralamayı görüyoruz. Bu TC numaralarına ait olan kişilerin çekilmesi için kullandığımız kod satırında dosyaları bu belirlenen sıra ile değil de , kendisine göre en küçük olan sayıları sıralayarak başlıyor.

6. Kaynakça

- [1] <https://www.kodlamamerkezi.com/c-net/c-ile-dosya-okuma-ve-yazma-islemleri/>
- [2] <https://www.kemalkefeli.com.tr/csharp-dosya-islemleri.html>
- [3] <https://enginpolat.com/csharp-binary-search-tree-ornegi/>
- [4] <https://github.com/yavuzselimfatihoglu/dataStructures>