

PROJE 3

Egemen Çakır

Kocaeli Bilgisayar Mühendisliği

Email: cakiregemen0@gmail.com

I. GIRIS

Bu projenin amacı, verilecek olan excel dosyalarında bulunan kişi bilgilerini kullanarak soy ağacı oluşturmak ve bu soy ağacı üzerinden çeşitli işlemler yapmaktır. Bu işlemler:

- Çocuğu olmayan düğümlerin listesinin yaş sıralamasına göre kaydedilmesi istenmektedir. Kaydetme adımında yapılan sıralama (sort) işleminin adım adım gösterilmesi beklenmektedir. (Depth First Algoritması kullanılacaktır.)
- Üvey kardeşler bulunarak harf sıralamasına göre kaydedilecektir. Kaydetme adımında yapılan sıralama (sort) işleminin adım adım gösterilmesi beklenmektedir. (Breath First Algoritması kullanılacaktır.)
- Kan grubu A olanların listesi kaydedilerek gösterilecektir. (Sunum esnasında kan grubu değiştirilebilir.)
- Soyunda aynı mesleği yapan çocuklar veya torunlar gösterilecektir. (Baba mesleğini, dede mesleğini devam ettiren kişiler ve mesleğini devam ettirdiği ataları)
- Soy ağacında aynı isme sahip kişilerin ismi ve yaşları gösterilecektir.
- Kullanıcıdan alınacak 2 tane isim girdisinden sonra büyük olan kişinin küçük olan kişiye yakınlığı gösterilecektir. (Örnek: Kişinin annesinin annesinin babası gibi.)
- Kullanıcıdan alınan kişi bilgisi ile o kişiye ait soy ağacının gösterilmesi istenmektedir. Verilmiş olan excel dosyalarına ek olarak kişinin soy ağacında bulunanların kişi ile yakınlık derecelerinin yazılması beklenmektedir. (Örnek: Kişinin annesinin annesinin babası gibi.)
- Soy ağacının kaç nesilden oluştuğu bulunacaktır. (Ağacın maksimum derinliği bulunacaktır.)
- Kullanıcıdan alınan isim girdisinden sonra o isimden sonra kaç nesil geldiği bulunacaktır

Bu proje javadan dosyadan veri alma ve işleme işlemlerini geliştirme açısından önemli ayrıca bu proje veri yapılarını daha fazla aşına olmamızı ve daha iyi kavramamızın yanında ağac veri yapısını kullanarak birçok işlem gerçekleştirip yukarıdaki isterleri gerçekleştirebilmeyi sağlaması açısından önemlidir. Bu projeyi java dilinde yazdım. Birçok sınıf kullandım , sınıflarda kapsülleme , miras alma gibi birçok yöntem kullandım. Ayrıca soy ağacını görsel bir şekilde gösterebilmek için swing'i ve birçok yararlı kütüphane kullandım.

II. ADIMLAR

Adım 1

Öncelikle verilen excel dosyasındaki 4 sayfada yer alan kişi bilgilerini dosyadan almam gerekiyordu bunun için Apache

POI(Apache POI, Word, PowerPoint ve Excel gibi Microsoft Office biçimlerinde dosyaları okumak ve yazmak için saf Java kitaplıkları sağlar.)'yi kullanarak dosyadan okuma işlemlerini gerçekleştirmeye çalıştım. Dosyadan verileri çekmek için izlediğim yol:

- Bir tane XSSWorkbook'dan dosya adında yeni bir nesne oluşturdum ve parametre olarak.xlsx dosyamın yolunu attım.
- XSSFSheet 'den sayfa adında bir nesne oluşturdum fakat bu sayfa dosya nesnesinden getSheetAt() metodu kullanılarak okunacak dosyanın istediğim sayfasına eşitledim.
- sayfa nesnesinde iterator() metodunu kullanarak itr adında bir iterator oluşturdum bu sayede sayfadaki satırlarda gezinebileceğim.
- Daha sonra bir while içinde Row row =itr.next() diyerek ilk satırımı aldım ve bir while döngüsü açıp o satırdaki tüm hücreleri gidip bilgileri aldım.

Bu bilgileri almayı başarınca bir Kisi sınıfı oluşturdum ve içinde string veri tipinde kişinin tc no (id), ad, soyadı, doğum tarihi, anne adı, baba adı, kan grubu ve mesleği, kızlık soyadı ve cinsiyet adında değişkenler koydum ayrıca daha sonra birkaç birşey daha ekledim. En son olarak hücreden çektiğim bilgiyi bir ArrayListe atıp bir satır okunduktan sonra tek tek 12 hücre bilgisini kişi sınıfındaki bilgiye eşitleyip Kisi tipinde nesneler tutan list adındaki arrayliste attım ve böylece dosyadaki her satır için bir kişi oluşturup depolayabildim.

Adım 2

Bu adımda projenin giriş ekranı için bir Giriş adında bir sınıf oluşturdum. Bu sınıfı JFrame'den extends ettim. Adım birde dosyadan okuma işlemlerini sürekli yazmamak için oku adında bir fonksiyonun içinde tekrar yazdım. Bu fonksiyon void dönüş tipinde ve parametre olarak bir tane int tipinde sayı ki bunu kaçinci sayfadaki bilgileri çekeceğimi belirlemek için kullanıcam ve iki tanede ArrayList alıyor. Bu sınıftaki kodlarımı sınıfın constructor'ı içinde yazdım. Constructorda ilk önce sayfa boyutunu ayrılayıp bir Panel oluşturup ekledim. Daha sonra panelin üstüne JLabel koyup içine başlık yazdım. Daha sonra 5 tane buton oluşturdum ve bunları yerleştirdim. Bu butonların amacı 4 sayfa için 4 aile mantığında her sayfada bizden istenen istrleri gösterdiğim sayfaları açmak için koydum. 5. butonuda daha sonra anlatacağım. En son olarak 4 butona isimlerini yazdım ve action ekledim actionda sadece bir sonraki adımlarda oluşturacağım sınıfların nesnelerini oluşturdum ve bu sınıfta yazdığım ve neredeyse oluşturacağım tüm sınıflarda

kullandığım metodu ekledim. Bu metodun adı gizle() bu metod sadece mevcut pencereyi setVisible(false) yaparak gizler.

Adım 3

Bu adımda Agac1 adında bir sınıf oluşturdum. Bu sınıfta JFrame'den miras aldım ve kodlarını constructor içinde yazdım. öncelikler daha önce yazdığım oku ve gizle metodlarını ekledim. Daha sonra sayfanın boyutunu ayarlayıp bir panel koydum. Sayfa tasarımına geçmeden önce agac yapımı oluşturmak için bir Node adında bir sınıf açtım. Bu sınıf özellik olarak Kisi sınıfından kisi adında bir değişken, c1, c2, c3, c4, c5 ve es adında Node veri tipinde değişkenler tutuyor ayrıca Kisi sınıfından veri tutan çocuk adında bir ArrayList, int tipinde seviye ve height adında değişkenler koydum. İki adet constructor ekledim biri parametresiz bir diğeri parametre olarak Kisi sınıfından nesne alıyor aldığı nesneyi Node sınıfındaki kisi değişkenine atıyor ayrıca height'ı 1 olarak atıyor. Daha sonra ekle, max ve yükseklik adında metodları yazdım. Bu metodlar:

- ekle metodu: Bu metod parametre olarak Kisi sınıfından kisi adında bir nesne, Node tipinde agac adında bir değişken alıyor ve bir Node dönüyor. Bu fonksiyonda ilk koşul gelen agac null ise yeni bir Node oluşturulup içine parametre olarak gelen kisi değişkeni atıp returnle agacı döndürür. Eger null değilse agac.kisi.es.equals(kisi.ad) kosulu sağlanırsa kosula girer ve yeni bir node nesnesi oluşturulup kisi parametre olarak gönderilir. Daha sonra agac.es=yeni oluşturulan node atanır ve return agac döndürülür. Eger bu koşulda sağlanmazsa cocugu olma ihtimali oluyor bunun içinde kisi anne adı ve baba adı nın agac.kisi.ad ve agac.kisi.es adlarına esit olma durumu kontrol edilir ve eger kosul sağlanırsa agacın özelliklerinden c1, c2, c3, c4 ve c5 den hangisi null değilse o koşula girer ve örneğin o koşul içinde agac.c1=ekle(agac.c1, kisi) şeklinde bir kod ve ardından return agac kodu gelir. Eger cocugu değil ve çocuklarının çocukları olma ihtimali var bunuda özyineli şekilde 5 tane c1, c2, c3, c4 ve c5 için ekle() metodlarını çağırdım ve bu sayede kendi kendine kontrolleri sağlıyor ve eklem işlemi sonrasında nodeları bağlayabiliyor. Ayrıca bu fonksiyonda max ve yükseklik metodlarını kullanıp her bir düğümün height değerini her eklemede güncelledim.
- yükseklik metodu: Bu metod parametre olarak Node tipinde bir agac değişkeni alıyor ve dönüş tipi int tipindedir. Bu fonksiyon eger agac==null ise 0 değerini döndürür aksi takdirde agac.height değerini döndürür.
- max metodu: Bu metod 5 tane int tipinde değer alır ve 5'inden hangisi büyük ise onu döndürür. Bu metod ve yükseklik metodunu agacın derinliğini bulurken kullandım.

Adım 4

Agac yapımı oluşturduktan sonra Agac1 sınıfının tasarımını yapmaya başladım. Bunun için başta eklediğim panelin içine 5 panel ve 5 buton ekledim. Bu paneller isterlerin sonuçlarını

görsüntülemek için kullandım ayrıca butonlardan birini ana sayfaya gitmek için diğerlerini isterler için kullanıcıdan bilgi alıp eklediğim panellere yazmak için kullandım. Bunları yaptıktan sonra oku metodunu constructor içinde çağırdım ve parametresini 0 atadım ve daha sonra global olarak bir Node agac değişkeni tanımlayıp kisileri depoladığım listi kullanarak ilk kişiden itibaren ekle metoduna agac ve kişileri parametre olarak gönderdim ve ilk sayfa için agac yapımı oluşmuş oldu. Daha sonra isterlerden biri olan çocukları olmayan kişilerin yaş sıralamasına göre listelenmesini yapmak için bir leaf adında bir metod yazdım. Bu metod:

- leaf metodu: Bu metod Deep First Search yapısına uygun olarak çalışıyor yani en derindeki kişiye ulaşmadan onun atasına kontrol tapılmıyor. Bu metod parametre olarak Node tipinde agac adında bir değişken alıyor. Sırasıyla özyineli şekilde leaf(agac.c1), leaf(agac.c2), leaf(agac.c3), leaf(agac.c4), leaf(agac.c5) şeklinde art arda çağırıyoruz. Daha sonra eger agac değişkeninin c1, c2, c3, c4 ve c5 değerleri null'sa demekki cocugu yoktur. Daha sonra global olarak tanımladığım leafs adında ArrayListe eger leafs.size()==0 ekledim fakat leafs de eleman varsa yaş kontrolüne göre eklemem gerekiyor bunun için agac.kisi'nin dogum tarihini gün, ay, yıl olarak ayırdım ve bir for içinde leafs içindeki kişilerin yaşlarını da ayırıp karşılaştırma yaptım. Eger agac.kisi bir elemandan küçük ise elemanın önüne ekledim fakat döngü bittiginde leafs.size()'ın eleman sayısı 1 artmadıysa agac.kisi demekki leafs elemanlarından büyük bu yüzden leafs'ın sonuna ekledim.

Bu metodu yazdıktan sonra metodu çağırdım parametresini gönderdim ve leafs'i for-each döngüsü ile eklediğim panellerden birine yazdırdım.

Adım 5

Bu adımda isterlerden aynı kan grubuna üye olanları bulmak için eklediğim butonlardan birine action ekledim ve burada JOptionPane ile kullanıcıdan bir kan grubu aldım daha sonra kanG adında bir fonksiyon yazdım. Bu fonksiyon:

- kanG metodu: Bu metod parametre olarak Node tipinde agac değişkeni ve String tipinde kan değişkeni alıyor. İlk önce gelen kan değişkenini split() metodu ile ayırıp (+) veya (-) kısmından kurtuldum. Daha sonra agac.kisi.kanGrubu değişkenini split() ile ayırıp daha sonra bu iki değeri karşılaştırdım ve aynı iseler global olarak tanımladığım kanGrubu adındaki ArrayListe atadım.

Bu fonksiyonu yazdıktan sonra butonun action kısmında kan grubunu aldıktan sonra bu metodu çağırıp parametrelerini verdim ve daha sonra for-each ile kanGrubu ArrayListinin elemanlarını diğer panele ekledim tabi bu ekleme işlemi kullanıcı butona bastıktan sonra gerçekleşecektir.

Adım 6

Bu adımda agacın derinliğini ekranda göstermek için bir JLabel ekledim ve bu labelı setText() metodunun içine

agac.height degerini yazdırarak agacın derinligini yani neslini yazdırdım.Hemen bunun altına bir tane daha JLabel koyarak kullanıcıdan girilen ismin devam neslini yazdırmak istedim.Bunun için eklediğim butonların birine action verdim ve içinde kullanıcıdan bir isim alıyor daha sonra bu ismi kullanmak için devamNesil adında bir foksiyon yazdım.Bu fonksiyon:

- devamNesil metod:Bu metod Node tipinde kok adında bir değişken ve String tipinde ad adında bir değişken alıyor.Daha sonra kok.kisi.ad ile parametre olarak gelen ad'ı karşılaştırıyor eger aynı ise global olarak tanımladığım int tipinde x adındaki değişkene kok.height degerini atıyor.Eger degile özyineli şekilde tüm Node'ları dolaşp kişi varsa onun derinligini x'e eşitliyor.

Daha sonra setText() ile x'i ekrana bastırıyorum.Bu isterden sonra aynı isme sahip kişileri eklediğim panellerden birinde göstermek için iki metod yazım bunlardan birini Node sınıfında digerini Agac1 sınıfında yazdım.Bu metodlar:

- isimler metodu:Bu metod Node sınıfının içinde ve dönüş tipi Node degerindedir.Parametre olarak Node tipinde agac adında değişken ve ArrayList alıyor.Bu fonksiyon özyineli şekilde agac.kisi'yi paramte olarak gelen ArrayListe ekliyor.
- aynıİsim metodu:Bu metod Node tipinde agac adında değişken alır ve dönüş tipi voiddir.Bu metodta öncelikler Node sınıfında yazdığım isimler metodunu çağırdım ve oradan bi kisi adında ArrayList elde ettim.Sonra bir while içinde bu kisi ArrayListinin eleman sayısı 0'dan büyük olana kadar devam edecek işlemleri yazdım.Bu işlemler öncelikler bir liste adında ArrayList tanımladım ve içine kisi ArrayListinin ilk elemanını attım daha sonra bir for döngüsü içinde kisi arrayinin ikinci elemanından başlayarak kontrol ettim eger liste içindeki elamanın adı ile liste ArrayListindeki mevcut elamanın adları aynı ise liste ArrayListine ekledim ve for bittikten sonra listenin eleman sayısı 2'den büyükse daha önce tanımladığım benzer ArrayListine eleman olarak attım ve kisi arrayinden kisi.removeAll(liste) diyerek kişileri sildim fakat tek eleman ise kisi.remove(0) diyerek mevcut kisiyi sildim ve while'ın kosulu gerçekleşmeyene kadar devam ederek aynı isimli arrayListleri eleman olarak benzer arrayListinde tuttum.

Bu iki metodu yazdıktan sonra yaptığım kod aşağıdadır:

```
aynıİsim(agac);
ArrayList<JLabel> lb13 = new ArrayList<JLabel>();
for(int i=0;i<15;i++)
{
    JLabel l =new JLabel("");
    lb13.add(l);
}
int syc3=0;
for(ArrayList<Kisi> k : benzer)
{
    if(k.size() >=2)
    {
        for(Kisi kisi : k)
        {
            p4.add(lb13.get(syc3));
            lb13.get(syc3).setText(kisi.ad+" id:"+kisi.id);
            syc3++;
        }
    }
}
```

Daha sonra aynı ata kişinin meslegini yapan kişileri listelemek isterini yapmak için iki metod yazdım bunlardan biri Node sınıfında birini Agac1 sınıfında yazdım.Bu metodlar:

- meslekler metodu:Bu metodu Node sınıfında yazdım.Node dönüş tipinde ve parametre olarak Node tipinde agac adında değişken ,tmp adında ArrayList ve meslek adında String bir değişken alıyor.Bu metod aldığı meslegi özyineli şekilde tüm nodların meslekleri ile karşılaştırıp tmp'e ekliyor.
- aynıMeslek metodu:Bu metodu Agac1 sınıfında yazdım.Void dönüş tipinde ve parametre olarak Node tipinde agac adında değişken alıyor.Öncelikle String meslek=agac.kisi.meslek diyerek ata'nın meslegini alıyoruz ve meslekler metodunu çağırıp paramtrelerini atıyoruz.

Bu iki metodu yazdıktan sonra for-each ile ekledigimiz panellerden birine ata mesegini yapanları yazdırdım.

Adım 7

Bu adımda üvey çocukları Breath First Search algoritması mantığında listelemek için iki metod yazdım.Bunlar:

- BFS metodu:Bu metodu Node sınıfında yazdım ve parametre olarak agac Node tipinde ve n adında bir arrayList alıyor.Öncelikle bu fonksiyon Nodları kok.seviye sıralamalarına göre n arrayListine ekiliyor.Ve bu Nodları dolaşma işlemi özyineli şekilde oluyor.
- uveyCocuk metodu:Bu metodu Agac1 içinde yazdım ve agac adında Node tipinde bir değişken alıyor.Daha sonra for(Kisi k : agac.cocuk) diyerek agacın cocuklarının üveylik durumları kontrol ediliyor.Bu işlem çocuk erkek ise soyismi atasınıniki ile aynı degilse global olarak tanımlanmış uvey arraylisti boş ise ekleniyor fakat boş degilse uvey ArrayListinin içindekiler ile harf önceligine göre uvey arrayListine ekleniyor.Cocuk erkek degilse bu sefer iki kontrol oluyor eger evlyise atasının soyismi ile kızlık soyismi kontrol ediliyor farklı ise yukarıdaki karşıştırmalı ve kontrollerden sonra ekleme işlemi oluyor.Fakat kadın evli degilse bu sefer ata soyadı ile kendi soyadı karşılaştırılıyor ve yukarıdaki kontroller ve ekleme yapılıyor.

Bu iki metodu yazdıktan sonra constructor içinde ilk önce BFS metodunu çağırırım daha sonra for-each ile uveyCocuk metoduna parametre olarak BFS den elde ettiğim Node'ları gönderdim ve panele üvey çocukları yazdırdım.

```
n.BFS(agac,node);
for(Node x : node)
{
    uveyCocuk(x);
}
int sync5=0;
ArrayList<JLabel> lb15 =new ArrayList<JLabel>()
for(int i=0;i<uvey.size();i++)
{
    JLabel l=new JLabel("");
    lb15.add(l);
}
for(Kisi k : uvey)
{
    p6.add(lb15.get(sync5));
    lb15.get(sync5).setText(k.ad);
    sync5++;
}
```

Adım 8

Bu adımda iki kişinin yakınlık değerlerini gösterme işlemini yaptım.İlk önce eklediğim butonlardan birine action verdim.Burada JOptionPane ile kullanıdan bir isim aldım ve daha önce yazdığım arama metodunu çağırırım.Arama metodunda eger birden fazla kişi bulunduysa kullanıcıda kişinin Id numarasını istedim daha sonra for each ile arama metoduna gönderdiğim arraylistin içindeki kişilerin id lerini karşılaştırıp eşleşen kişiyi yeni bir tmp adında arrayListe attım daha sonra kullanıcıdan ikinci kişiyi aldım ve yukarıdaki işlemler uygulandı artık tmp arrayListimizde iki kişi var bundan sonrası için iki metod yazdım.Bunlar:

- akrabalık metodu: Bu metod tmp adında ArrayList alıyor ayrıca int tipinde bir deger alıyor.İlk önce tmp içindeki iki kişinin büyük küçük olarak belirledim.Sonra ilk koşulumuz olan eger p=k1.seviye-k2.seviye burdan çıkan sonuc p==0 ise üç ihtimal vardır.İlki ikisinin anne baba adları aynı ise ekrana "kardeş" yazdırdım eger anne baba adları farklı fakat birbirlerinin eşleri iseler "eşidir" yazdırdım son olarak ikisinde değilse ekrana "kuzen" yazdırdım.Diğer koşulumuz p==1 veya p== -1 durumu bu koşulda da "anne","baba","dayı","teyze","amca","hala" olma durumlarını kontrol ettirdim.Eger bu iki koşul sağlanmazsa ilişki metodunu çağırırım ve oradan gelen sonucu ekrana JOptionPane ile gösterdim.
- ilişki metodu:kucuk,buyuk adında iki tane Kisi tipinde değişken alıyor ayrıca x adında String tutan bir arrayList alıyor.İlk önce kucu.ata!=null olmama koşulunda eger kucuk.ata.e!=null değilse burada "annesi","babası" olma durumuna göre x.add() diyerek ekliyoruz.Eger kucuk.ata.cocuk.size()!=0 ise for-each ile "kız kardeş","erkek kardeş" kontrolü yapıp x'e ekleniyor.Eger hiçbirisi gerçekleşmez ise Kadın veya Erkek olma durumuna göre x'e "babasının" ya da "annesinin"

eklenip özyineli olarak ilişki(kucuk.ata.kisi,buyuk,x) şeklinde devam ediyor.

Bu fonksiyonları yazdıktan sonra tmp.size()==2 koşulunu sağlama durumunda çalışıyoe ve JOptionPane ile ekrana akraba ilişkilerini gösteriyor.

Adım 9

Bu adımda boşta olan son butonuma action ekledim.Bu butonun amacı bu ailenin soy ağacını ekranda göstermek onun için soy1 adında bir sınıf oluşturdum ve bu action içinde bu sınıftan nesne oluşturdum.soy1 sınıfı:

- soy1 sınıfı: Bu sınıf JFrame'den miras alıyor oku,kontrol,yapı2 metodlarını içeriyor ve bütün kodları constructor içinde yazdım.Metotları: *oku metodu:Agac1 sınıfındaki metod ile aynı.
*yapı2 metodu:Bu metod agactaki kişilerin isimlerini butonlara yazıp ekranda belli mesafelerde yerleştirip soy ağacı görüntüsü veriyor.
*kontrol metodu:agac adında Node tipinde bir değişken ve k adında Kisi tipinde bir parametre alıp eşi olup olmadığının kontrolünü yapıp global olarak tanımlanmış sync değişkeninin değerini değiştiriyor.Bu metodu yapı2 içinde kullandım.

Böylece ilk sayfa için soy ağacım ve isterleri yapmış oldum.Giris sayfasındaki diğer Agac2,Agac3,Agac4 butonları için Agac1 gibi yeni bir sınıf açtım ve sadece Agac1'den farklı olarak oku() metodunun parametrelerini 1,2,3 olarak her sınıf için değiştirdim.

Adım 10

Bu son adımda Giris sınıfında yer alan son butona action verdim.Bu butonun amacı kullanıcıdan girilen bir kişinin soy ağacını ekrana getirmek onun için önceden eklediğim oku metodunu 4 sayfa için çağırırım ve her sayfa için list,list3,list5,list7 kisi arrayListlerimin yanında agac1,agac2,agac3 ve agac4 Node tipinde global değişkenlere sahip oldum.Action içinde ilk önce kullanıcıdan bir isim aldım ve Node içindeki arama fonksiyonunu her sayfa için çağırırım ve bir kişi adında kişi tipinde degerler tutan ArrayListe attım eger kisi arrayListinde 0 eleman varsa ekrana "hatalı giriş yazdırdım",eger 1 kişi varsa demekki Id istemeye gerek yok bu yüzden bu koşul içinde iki koşul yazdım eger atası yoksa demekki bu kişi atadır o yüzden tüm sınıfları ata isimleri ve id lerini karşılaştırıp hangi koşul gerçekleşirse soy1 sınıfından nesne üretip parametre olarak o sayfanın numarasını gönderdim ve ekrana o kişinin bulunduğu soy ağacı geldi.Eger kisi arrayListinin elema sayısı 2 ve 2'den büyükse bu sefer Id istedim ve for döngüsü ile kisi'nin elemanlarını gezip kontrol sağladım ve elemanı bulduğumda eleman sayısı 1 olma durumunda gerçekleştirdiğim adımları uyguladım.Bu işlemide yaptıktan sonra projeyi tamamlamış oldum.

III. İSTERLERİN EN KÖTÜ - EN İYİ KARMAŞIKLIK HESAPLARI

- 1. İsterin Karmaşıklik Hesabı:

IV. ÖZET

Bu projenin amacı veri yapılarını kütüphane kullanmadan dosyadaki verileri okuyarak bir soy ağacı çıkarmaktır. Bu projede kendime veri yapıları ile ilgili birçok şey kattım ve java ile ilgili farklı tipte bir dosyadan veri çekip kullanma işlemini gerçekleştirip öğrendim. Bu projenin sonunda istenen bütün isterleri yerine getirmeye çalıştım.

V. SONUÇ

Bu projede kontroller yapıldıktan sonra bir ana ekranım var ve bu ekranda aileleri seçip onlarla ilgili isterleri ekranda gösterebiliyorum fakat belli bir kişinin soy ağacının gösterilmesi veya ailenin soy ağacının gösterilmesi konusunda çok fazla sıkıntı çektim. Bu sorunlar:

- 1.Durum: Belli nesilden sonra butonların iç içe geçme sorunu bu soruna çözüm bulamadım çünkü sayfa boyutunu max seviyede kullandığım halde bu sorun çıkıyor.
- 2.Durum: Bu projenin görsel kısmını swing ile yaptığım için ailelerin çocukları ile bağlantılarını gösteremedim.
- 3.Durum: Soy ağacını çıkartırken sadece soy ağacında mevcut kişilerin adlarını butona ekleyip onların ekranda gösterdim.

VI. KAYNAKÇA

- <https://poi.apache.org>
- <https://www.javatpoint.com/java-swing>
- <https://www.w3schools.com/java/default.asp>
- <https://www.javatpoint.com/java-oops-concepts>