İNSAN KAYNAKLARI VE KARİYER BİLGİ SİSTEMİ

VERİ YAPILARI

PROJE-3

**Ali Yeşilkanat** – 05120000223  
**Gökçe Yardımcı –** 05110000049  
**İzel Çavuşoğlu –** 05120001203

**İçindekiler**

İçindekiler

Program 3A

1. Programcı Kataloğu

1.1.Gerçekleştirilen Platform

1.2. Proje Tanımı  
1.3. Kullanılan sınıfların ve metotlarının açıklamaları ....................................................  
1.4. Kullanılan veri yapıları ve açıklamaları.....................................................................  
1.5. Kullanılan dosyaların özellikleri ve açıklamaları....................................................  
1.6. Yazılım geliştirme için harcanan süreler................................................................

2. Kullanıcı Kataloğu

2.1. Programın İşletimi ve Ekran Görüntüsü

2.2. Kullanıcı Kılavuzu

2.3. Programın Kısıtlamaları

**1.PROGRAMCI KATALOĞU**

**1.1. Gerçekleştirilen Platform:** Visual Studio 2010 , .Net Framework 4

**1.2.Proje Tanımı:** Geliştirilecek bir insan kaynakları ve kariyer bilgi sistemi kapsamında, iş başvurusu yapan kişilere ilişkin bilgiler, kişi adına göre, ikili arama ağacı yapısı üzerinde tutan ve ağacın her düğümü, bir kişiye ilişkin bilgileri (adı, adresi, telefon, e-posta, uyruğu, doğum yeri, doğum tarihi, medeni durum, yabancı dil, ilgi alanları, referans olan kişiler) içeren programı yazmamız istenmektedir. Ayrıca her bir düğümde, o kişinin “İş Deneyimi” ve “Eğitim Durumu”birer dinamik listede tutulacaktır.

Eleman arayan şirketler de, bilgilerini (işyeri adı, tam adresi, telefon, faks, e-posta) ve iş ilanlarını sisteme yükleyebileceklerdir. İşyeri bilgileri ve başvuru yapan kişiler, işyeri adınagöre bir Hash Table içerisinde tutulacaktır. Hash Table‘da her bir iş ilanı için bir Heap (ağaç şeklindeki öncelik kuyruğu) tutulacaktır. Heap’in her bir düğümü, o şirketin ilgili iş ilanına başvuru yapan kişinin bilgilerinin referansını ve o kişinin işe uygunluğunu gösteren bir sayıyı içerecektir. Bu sayı 0.0 ile 10.0 (double) arasında olup, kullanıcı tarafından başvuru sırasında girilebilecek veya istendiğinde program tarafından rastgele olarak da üretilebilecektir; 10.0 değeri, o işe en uygun eleman olduğunugöstermektedir (0.0 ise hiç uygun olmadığını).

Bu bilgilere göre aşağıdaki bölümlerden oluşan programı geliştirmemiz istenmektedir.

1. İş Başvurusu Yapan Kişilerin Kullanacağı Bölüm
2. Eleman Arayan Şirketlerin Kullanacağı Bölüm
3. Ağaç İşlemleri

**1.3. Kullanılan Sınıfların Ve Metotlarının Açıklamaları:**

**Program.cs**

publicstaticint menu()

Programın ana menüsünü ekrana yazdırır.

publicstaticint altMenu(int ustMenuSecim)

Ana menüden seçim yapıldığında seçime göre ekrana gelecek alt menüleri ekrana yazdırır.

privatestaticvoid YeniElemanEkle(Tree elemanlar)

İş başvurusu yapan kişilerin sisteme kaydı yapılır, kendi kişisel bilgileri girilir.

privatestaticvoid YeniSirketEkle(Hashtable sirketler)

Eleman arayan şirketlerin sisteme kaydı yapılır, işyeri bilgileri eklenir.

privatestaticvoid KisininBilgileriniGuncelleme(TreeNode nod)

İş başvurusu yapan kişilerin bilgilerini günceller. Değiştirmek istenilen bilginin indeksi girilerek istenilen bilgilerde değişiklikler yapılır ve güncelleme işlemi tamamlanır.

privatestaticvoid SirketGuncelleme(Hashtable sirketler)

Eleman arayan şirketlerin bilgilerini günceller. Değiştirmek istenilen bilginin indeksi girilerek istenilen bilgilerde gerekli değişiklikler yapılarak güncelleme işlemi yapılır.

privatestaticvoid IlaniSistemdenKaldir(Hashtable sirketler)

İş ilanını kimseyi işe almadan çekmek için kullanılır.

privatestaticvoid UygunAdayiIseAl(Hashtable sirketler)

İş ilanı için en uygun aday elemanların işe uygun adayın işe alınmasını sağlar.

privatestaticvoid KisiListele(Tree elemanlar)

Ağaç işlemlerinde adından kişi arama yapar ve kişinin tüm bilgilerini listeler.

privatestaticvoid IseBasvuranAdaylariGoruntule(Hashtable sirketler)

İşe başvuran tüm adayların bilgilerini listeler.

privatestaticvoid IsIlaninaBasvurma(Hashtable sirketler, Tree elemanlar)

Sistemdeki bir işe başvurmak için kullanılır.

privatestaticvoid ElemanOku(Tree elemanlar)

eleman.txt dosyasından veriler alındı.

privatestaticvoid SirketleriOku(Hashtable sirketler)

sirket.txt dosyasından veriler alındı.

privatestaticint SayiAl()

privatestaticint SayiAl(int a, int u)

privatestaticdouble SayiAl(double a, double u)

Sayıların kullanıcıdan alınmasını sağlar. Eğer kullanıcı sayıdan başka bir girdi girerse kullanıcıya hata mesajı verip tekrar girmesini ister. Argüman isteyen aşırı yüklenmiş versiyonu ise, kullanıcının üst ve alt sınır değerleri arasında bir sayı girmesini sağlar.

**EgitimDurumu.cs**

Kişilerin not ortalamalarını,okullarını,okullarının başlangıç ve bitiş tarihlerini, bölümlerini tutan sınıftır.

**Eleman.cs**

public void egitimleriListele(): Kişilere ait eğitim durumlarını listerler

public void isDeneyimleriniListele():Kişilere ait iş deneyimlerini listerler.

**Heap.cs**

 public void kisiyiIlandanCikar(Eleman e): İşe giren bir kişiyi sistemdeki tüm iş ilanlarından çıkarır.

 public Boolean isEmpty(): Heapin boş mu dolu mu olduğunu kontrol eder.

 public void trickleUp(int index): Yerleştirilmek istenen metodunun yerinin bulunması için önemlidir.

public HeapNode remove(): Heapden elemanı siler ve sildiği elemanı döndürür.

public void trickleDown(int index):Heapden eleman silmek için gereklidir.Eleman silindikten sonra ortaya çıkacak olan yeni düzenleme için gereklidir.

 public void displayHeap(): Heapdeki kişileri ekrana yazdırır.

 public Boolean insert(double key, Eleman e): Heape yeni bir eleman ekler.

 public HeapNode ilaniSistemdenCikar(): En uygun elemanı alınmış olan iş ilanını sistemden çeker.

**HeapNode.cs**

Bir heap düğümünü oluşturan class.

**IsDeneyimi.cs**

Kişilerin önceden çalıştıkları şirketlerin isimlerini,adreslerini ve orada çalıştıkları pozisyonları tutar

**Sirket.cs**

Şirketlerin adlarını, adreslerini,telefon ve faks numaralarını,epostalarını ve verdikleri iş ilanlarını tutar.

public override string ToString(): Şirketin özellliklerini göstermek için yazılmıştır.

**Tree.cs**

 public void notOrt90Uzeri(TreeNode etkin): Not ortalaması 90ın üzerinde olan kişileri yazdırır.

 public void derinlikVeDugumSayisi(TreeNode etkin):Derinlik ver düğüm sayısını bulup,ekrarna yazdırır.

public void ingilizceBilenler(TreeNode etkin):İngilizce bilen kişileri ekrana yazdırır.

 public TreeNode ara(TreeNode localRoot, string kisi):İsmi verilen kişiyi arar.

public void preOrder(TreeNode localRoot, int d): Ağacı preorder düzeninde dolaşır

 public void inOrder(TreeNode localRoot, int d) Ağacı inorder düzeninde dolaşır

 public void postOrder(TreeNode localRoot, int d) Ağacı postorder düzeninde dolaşır

 public void insert(Eleman newEleman): Ağaca yeni bir eleman ekler.

 public void düzeyListele(TreeNode etkin, int d):Agacın düzeylerindeki elemanları listeler.

 public void düzeyListele2(TreeNode etkin): Agacın düzeylerindeki elemanları listeler.

 public bool delete(string key): Adı verilen kişiyi ağaçtan siler

 private TreeNode getSuccessor(TreeNode delNode): Silinen düğümden sonraki alfabetik sıraya göre sonraki düğümü bulur.

**TreeNode.cs**

Bir ağaç düğümünü oluşturan class.

**1.4. Kullanılan Veri Yapıları ve Açıklamaları**

**Eleman.cs**

isDeneyimleri = new List<IsDeneyimi>();

 egitimler = new List<EgitimDurumu>();

 basvurduguIsIlanlari = new List<Heap>(): Kişinin başvurduğu iş ilanlarını tutan liste.

**Heap.cs**

HeapNode[] oldHeapArray

HeapNode[] heapArray

**Sirket.cs**

 isIlanlari = new List<Heap>(50): İlana başvuran kişileri işe uygunluk değerlerine göre tutar.

**Program.cs**

Hashtable sirketler = new Hashtable():Şirketleri isimlerine göre tutar.

Tree elemanlar = new Tree(): Kişileri isimlerine göre sıralı tutar.

string[] elemanOzellikleri

string[] isDeneyimiOzellikleri :İlgili sınıftaki özelliğin ne olduğunu

string[] egitimOzellikleri tutan string dizileri.Örn telefon,eposta..

string[] sirketOzellikleri

string[] elemanBilgileri

string[] isDeneyimiBilgileri : Yukarıdaki dizilerdeki özelliklere

string[] egitimBilgileri karşılık gelen değerleri tutar.

string[] sirketBilgileri örn: 055234524, asdf@hotmail.com

List<Heap> sistemdekiAktifIlanlar = new List<Heap>(): Sistemdeki ilanları tutar.

**1.5. Kullanılan Dosyaların Özellikleri ve Açıklamaları**

**Dosyalar hakkında genel özellikler:**

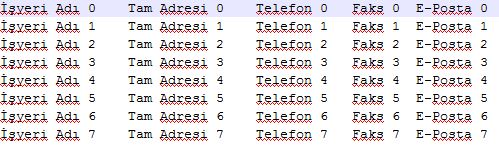
1. Dosyalar proje dosyasının içinde bin\debug klasörü içinde yer almalıdır.
2. Dosyalardaki her bir satır ait olduğu sınıfın sahalarının verilierini içerir.
3. Dosyalardaki veriler \t (TAB) ile birbirilerinden ayrılmıştır.
4. <http://yzgrafik.ege.edu.tr/~tekrei/dersler/veriyapilari/dosyalar.zip> adresindeki dosyalar kullanılmıştır.

**Şirketler Dosyası (sirket.txt):**

Şirketlerin adlarının, adreslerinin, telefonlarının, faks ve e-postalarının tutulduğu metin dosyasıdır.

“İşyeri Adı”, “Tam Adresi”, “Telefon”, “Faks”, “E-Posta” stringlerinden bir boşluk sonraki veri o sahanın verisidir. Bu stringlerde sirket.txt dosyasında \t ile birbirlerinden ayrılmıştır. Her bir satırda bir şirketin verileri tutulmaktadır.

Programda yukarıdaki stringler, String.IndexOf metodu ile başladıkları indeks bulunup, satırdaki istenilen yere getirilip \t görülene kadar okunup ilgili değişkene atılmıştır.



**Elemanlar Dosyası (eleman.txt):**

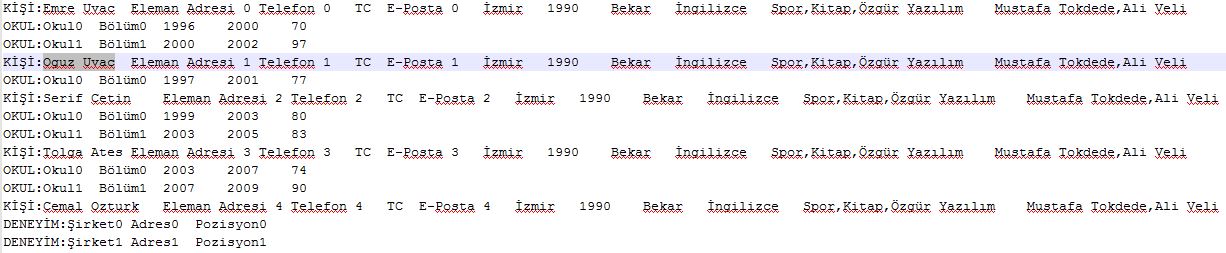
İşlere başvuracak elemanların kişisel bilgilerinin, geçmişte çalıştığı işlerin ve mezun olduğu okulların tutulduğu metin dosyasıdır.

Satır “KİŞİ:” ile başlıyorsa, yeni bir eleman nesnesi oluşturulup sonraki “KİŞİ:” ile başlayan satıra kadar okunan bilgiler o nesne içine atılır. Satır “OKUL:” ile başlıyorsa, satır okunup kişinin bilgilerinin ait olduğu eleman nesnesinin içindeki mezun olduğu okulları tutan veri yapısına atılır. Satır “DENEYİM:” ile başlıyorsa mezun olduğu okulların aynı durumu geçerli olup, o satırdaki bilgileri ilgili eleman nesnesinin içindeki geçmiş iş deneyimlerini tutan veri yapısına atılır.

Okul bilgilerinin okunmasında “Okul” ve “Bölüm” stringlerinin bir sağındakinden \t’ye kadar olan string o veriyi verir. Sonrasındaki tüm okul bilgileri \t ile ayrılmıştır.

Deneyim bilgilerinin okunmasında ise “Şirket”, “Adres” ve “Pozisyon” stringlerinin sağındakinden \t’ye kadar olan string o veriyi verir.

Kişisel Bilgilerin okunmasında ise kişinin ismi ile başlanır. “Eleman Adresi”, “Telefon”, “E-Posta” ile başlayan stringlerden bir boşluk sonrasından başlayan \t’ye kadar olan veri o bilgiyi belirtir. Diğer tüm verilir \t ile ayrılmıştır sırasyıla okunmak zorundadır.

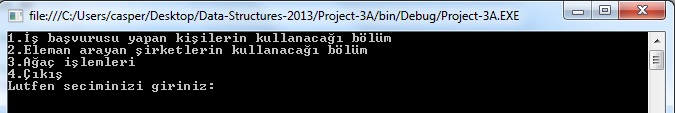


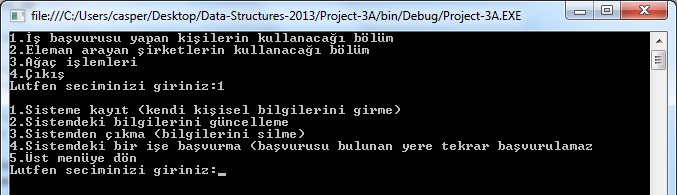
**1.6. Yazılım Geliştirme İçin Harcanan Süreler**

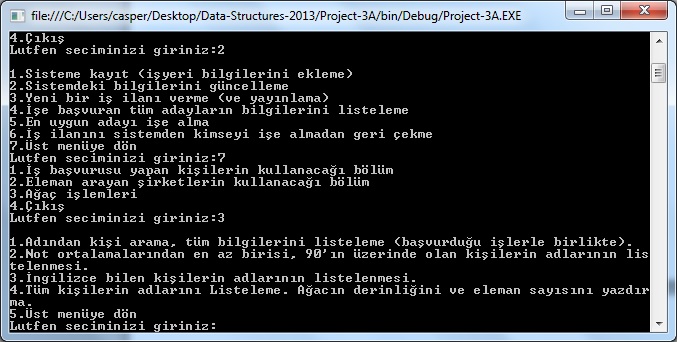
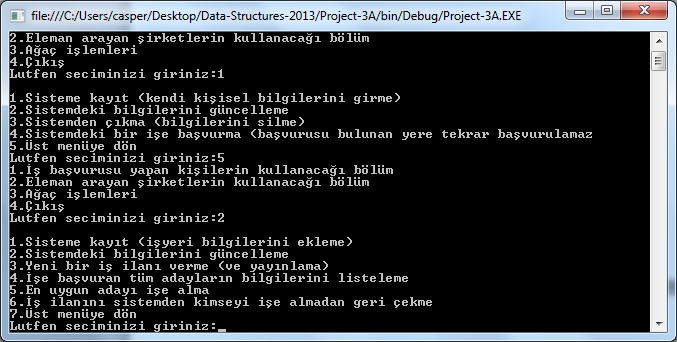
Ali Yeşilkanat 15 saat  
Gökçe Yardımcı 6 saat  
İzel Çavuşoğlu 4 saat

**2.KULLANICI KATALOĞU**

**2.1.Programın İşletimi ve Ekran Görüntüsü:**

****

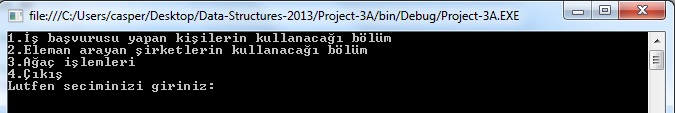
****

****

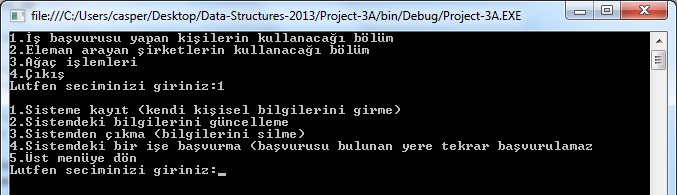
**2.2.Kullanıcı Kılavuzu:**

**1.İş Başvurusu Yapan Kişilerin Kullanacağı Bölüm**

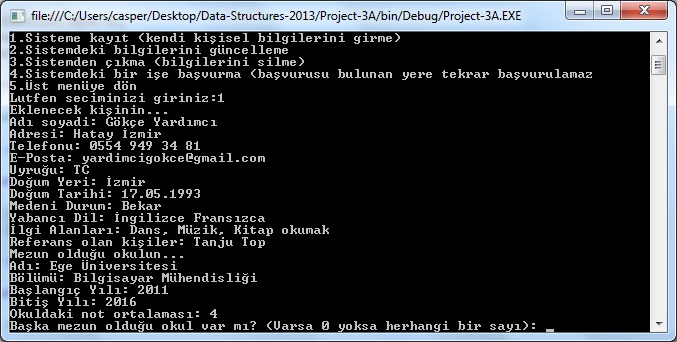
1) Program çalıştığında aşağıdaki ekran karşınıza gelmektedir.

****

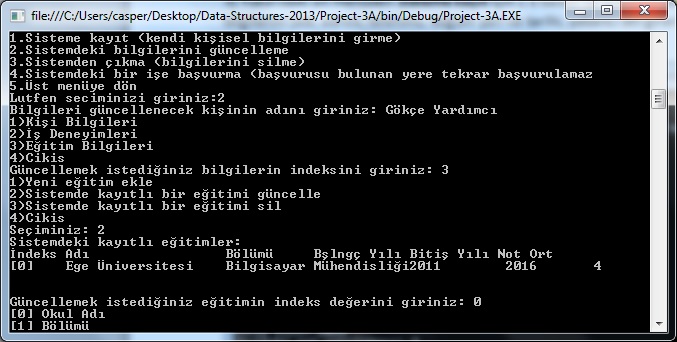
2) Seçim olarak 1 girildiğinde **İş başvurusu yapan kişilerin kullanacağı bölüm** seçilerek aşağıdaki alt menünün görüntülenmesi sağlanır.



3) Yukarıdaki alt menüden **Sisteme kayıt** için 1’e basılır ve iş başvurusu yapan kişilerin kendi bilgilerini(adı, adresi, telefon, e-posta, uyruğu, doğum yeri, doğum tarihi, medeni durum, yabancı dil, ilgi alanları, referans olan kişiler)girerler. Ayrıca kişinin iş deneyimi ve eğitim bilgileri de girilir.



4) **Sistemdeki bilgileri güncellemek** için ise 2’ye basılır. Güncellenecek bilgilerin indeksi girilerek hangi bilginin güncelleneceği seçilir ve gerekli değişiklikler yapılır.

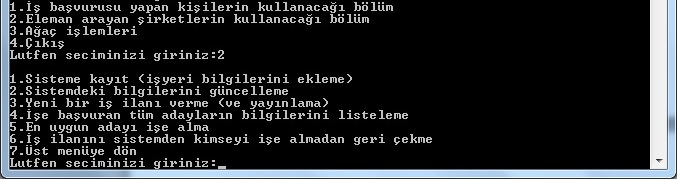


5) İş başvurusu yapan elemanların kendi bilgilerini silmesi için **Sistemden çıkma**yani 3’e basmaları gerekmektedir.

4)**Sistemdeki bir işe başvurmak için** 4’e basılır ve iş ilanlarından istenilenin indeksi seçilerek başvuru yapılır.

**2.Eleman Arayan Şirketlerin Kullanacağı Bölüm**

1) Eleman arayan şirketlerin kullanacağı bölüm için ana menüden 2’ye basılır ve aşağıdaki ekran görüntülenir. Alt menüden istenilen işlem seçilir ve buna uygun olarak işlemler yapılır.

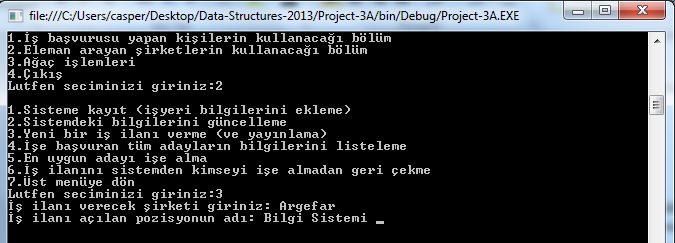


2) **Sisteme kayıt** ve **Sistemdeki bilgilerini güncelleme** “İş başvurusu yapan kişilerin kullanacağı bölüm”deki sisteme kayıt ve sistemdeki bilgilerini güncelleme ile benzer şekilde çalışmaktadır.

İşyeri bilgileri eklemek için **Sisteme Kayıt** seçilir yani 1’e basılır.

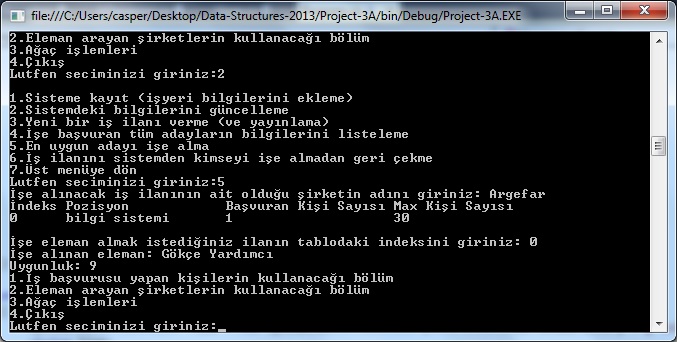
Sistemdeki işyeri bilgilerini güncellemek için ise **Sistemdeki bilgilerini güncelleme** seçilir yani 2’ye basılır.

3)**Yeni bir iş ilanı vermek** için 3’e basılır. İş ilanını veren şirketin adı ve pozisyonun adı girilir.

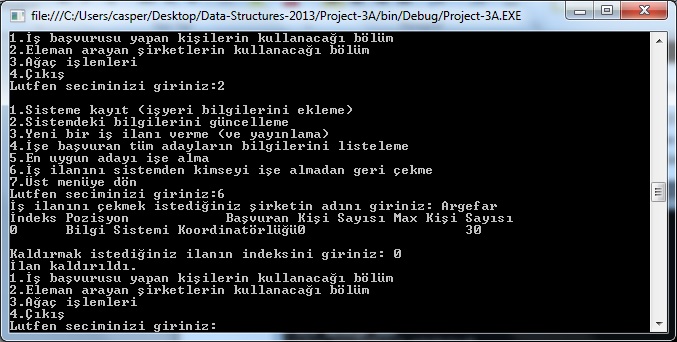


4) **İşe başvuran tüm adayların bilgilerini listelemek** için 4’e basılır.

5) Başvuran adaylardan iş için **en uygun adayı işe almak** için 5’e basılır. Hangi şirketin hangi iş ilanı için bu işlemi gerçekleştirmek istiyorsanız o ilan seçilir ve o iş ilanı için en uygun aday elemanların işe uygunluğunun sayısına göre belirlenerek görüntülenir.

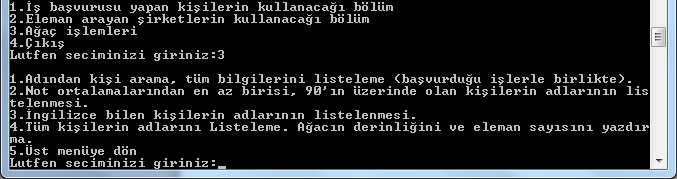


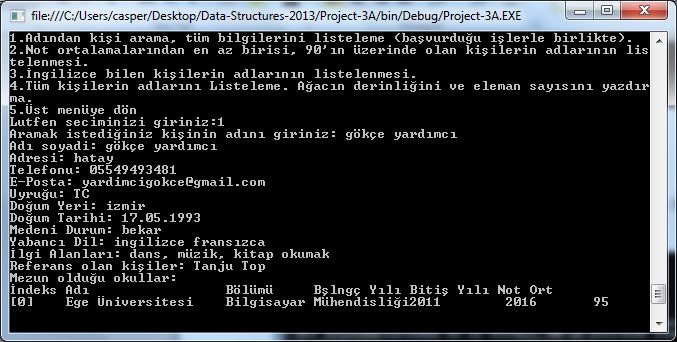
6) **İş ilanını sistemden kimseyi almadan geri çekmek** için 6’ya basılır. İş ilanını çekmek istediğiniz şirketin adını ve kaldırmak istediğiniz ilanın indeksini girerek bu işlem gerçekleştirilir.



**3. Ağaç İşlemleri**

1) İkili arama ağacı üzerinde işlem yapmak için ana menüden 3’e basılır ve aşağıdaki ekran görüntülenir.

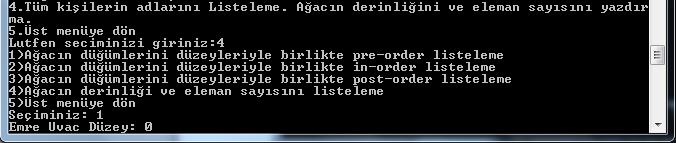


2) Ağaç İşlemleri’nin alt menüsünden 1’e basarak **adından kişi arama** yapılabilir ve o kişiye ait **tüm bilgiler** listelenebilir.

3) **Not ortalamalarından en az birisi, 90’ın üzerinde olan kişilerin adlarının listelenmesi** için 2’ye basılır.

4) **İngilizce bilen kişilerin adlarının listelenmesi** için 3’e basılır.

5) **İkili arama ağacındaki tüm kişilerin adlarını düzeyleri ile birlikte listelemek** için 4’e basılır.



**4. Çıkış**

Çıkış için ana menüde 4’e basılır ve programdan çıkış yapılır.

**2.3. Programın Kısıtlamaları**

* sirket.txt ve eleman.txt dosyaların bin ve debug adresinde olmalı
* Şirket ve kişi adı güncellenemez.
* Bir iş ilanına maksimum 30 kişi başvurabilir.