T.C.
EGE UNIVERSITESİ
MÜHENDISLIK FAKÜLTESİ
BILGISAYAR MÜHENDISLIK BÖLÜMÜ

VERİ YAPILARI DERS

Proje-3:

İNSAN KAYNAKLARI ve KARIYER BILGI SİSTEMİ

AD SOYADİ: Tsogtgerel CHIMEDTSEREN, Erdene JARGAL,

OGRENCİ NO: 05080010088, 05060008421

İZMİR.23/12/2009

1.1.Gerçekleştiren Platform- Dil ve Sürüm Adi:

Java, Eclipse EE

1.2.Problemin kısa tanımı:

Projede iş arayan her işçi ve işçi arayan her şirketler için ayrı ayrı bilgilerin tutululacağı ve istedikleri zaman güncelleme yapabileceği bilgi sistemin oluşturulması istendi. Program bilgi sistemi için ikiye bölünmektedir ve kim olduğuna göre kullanıcı başlangiçta mönü üzerinden seçimini yapıp ilgili mönüye geçer. İşçiye iliskin bilgiler adı, adresi, telefonu, e-posta, uyruğu, dogum yeri, doğum tarihi vs..., Şirkete ilişkin bilgiler işyeri adı, tam adresi, telefon, faks, e-posta kapsamındadır. Mönü üzerinde kullanıcı işçi ise bilgilerini sisteme ekleme,sistemden silme,sistemde güncelleme ve sirketlerinin ilanlarına başvurabilme, şirket ise bilgilerini ekleme, silme, günleme, yeni ilan verme ve ilana göre başvuran işçilerden belli kurala göre işe alma gibi işlemler yapabilmektedir.

1.3 Kullanılan sınıfların ve metotlarının kısa açıklamaları

KişiBilgi Package - *içinde olan sınıfları* :**IsciDugm, IsDeneyim,EgitimDurum, IsciBinaryTree.**

IsciDugum sınıfı - *İşçilerin bilgilerini tutmaktadır.bilgi sistem yapısı ağaç olduğundan sınıf kendi tipinde sağ ve sol düğümleri tutar.*

Metotları:

Bilgileri almak ve atamak için getter ve setter metotlar örneğin ad icin:

public void setAdi(String adi)

public String getAdi()

Kişinin bilgilerini görüntülemek için

public void displayIsciDugum()

IsDeneyim sınıfı - *Kişinin iş deneyimine ilişkin bilgileri tutar.*

Metotlar:

Bilgilerin sisteme alınması için getter metotlar. Bilginin girilmesi constructorla yapılır. örnek:

```
public String getIsyerAdi()
public IsDeneyimi(String isyerAdi, String isyerAdresi,
                  String isyerPozisyon, String isyerGorevi)
```

İş deneyimini görüntülemek için

```
public void displayIsDeneyim()
```

EgitimDurum sınıfı - *Kişinin eğitim durumuna ilişkin bilgileri tutar.*

Metotlar:

Bilgilerin sisteme alınması ve atanması için getter ve setter metotlar. örnek:

```
public String getOkulAdi()
public void setOkulAdi
```

İş deneyimini görüntülemek için

```
public void displayEgitim()
```

IsciBinaryTree sınıfı - *Ağaç yapısına sahip ve işçilerin bilgilerini tutan işçiduğumleri ağacın düğümü gibi tutar. Eleman olarak ağacın kök ve düzeyi tutar.*

Metotlar:

public IsciBinaryTree() {root=null;}
Ilkleme için constructur

public IsciDugum getRoot()
Kökü çağırmak için

public void setRoot(IsciDugum root)
Köke atama yapmak için

public boolean isEmpty()
Ağacın boş olup olmadığını control edip true false döndürür:

public void insert(IsciDugum newNode)
ekleme yaparken kolaylık sağlamak için yapılan bir metottur. Aşagdaki metoda ağacın kökü ve newNode ikisini parameter olarak verip çağırır.

private IsciDugum insert(IsciDugum newNode, IsciDugum current)
Ağaca dügümleri ekler. Eklerken ağacın dengesini bozmadan ekler. Burda AVL Tree yapısına gore ekleme yapılır.

private IsciDugum rotateWithLeftChild(IsciDugum k2)
localroot`un sol tarafı ağır oldugu zaman dengelemek için sağ tarafa çevirip sol çocuğu kök yapar.

private IsciDugum rotateWithRightChild(IsciDugum k1)
localroot`un sağ tarafı ağır oldugu zaman dengelemek için sol tarafa çevirip sağ çocuğu kök yapar.

private IsciDugum doubleWithLeftChild(IsciDugum k3)
Double rotate binary tree node: önce k3`ün sol çocuğunu sağa çevirip; sonra k3`ü yeni sol çocuğu ile çevirir.

private IsciDugum doubleWithRightChild(IsciDugum k1)*Double rotate binary tree node: önce k1`in sağ çocuğunu sola çevirip; sonra k1`i yeni sağ çocuğu ile çevirir.*

private IsciDugum getSuccessor(IsciDugum delNode)
silinecek düğümün yerine geçebilecek en uygun düğümü bulup gerekli bağlamakarın yapıldığında düğümü döndürür.

public void remove(String ad)
Adını anahtar olarak alıp eğer kişinin bilgi bulunursa düğümü silip ağacı dengeler. Bulunmadıysa bulunmadı mesaj verir.

public IsciDugum remove(String ad, IsciDugum node)
remove metodun alt metodudur. Silme ve dengeleme işlemi asıl gerçekleştiren metot.

public void balanceRoot()
ağaç dengesizse ağacı dengeler.

public void preorder(IsciDugum localRoot)
ağacı preorder dolaşıp kişinin adını düzeyile ekrana yansıtır.

public void inorder(IsciDugum localRoot)
ağacı inorder dolaşıp kişinin adını düzeyile ekrana yansıtır.

public void postorder(IsciDugum localRoot)
ağacı postorder dolaşıp kişinin adını düzeyile ekrana yansıtır.

public IsciDugum bul(String input)
Kolaylık için agactanArama() metodunu çağırır. Arama işlemini gerçekleştirip geri döndürür. Bulunmazsa bulunmadı mesaj verir.

private IsciDugum agactanArama(IsciDugum localroot, String input)
Ağaçta arama yapıp düğümü bulup döndürür. Bulunmazsa NULL döndürür.

private static int height(IsciDugum current)
Düğümün yüksekliğini sağ sol çocaga göre hesaplayıp yüksekliği döndürür.

public void ortalamaGoreListele(double ortalama,IsciDugum localRoot)
Ağacı dolaşıp ortalaması 90 üzerinde olanları listeler.

public void yabanciDilBilen(String dil,IsciDugum localRoot)
Yabancı dil bilenleri listeler.

Sirket Package – *içinde olan sınıflar:* **IsyerBilgi, IsIlan, Sirket**

IsyerBilgi sınıfı – *Şirketlerin bilgilerini tutar. Iş ilanları vector olarak tutar.*

Metotlar:

Bilgileri almak ve atamak için getter ve setter metotlar örneğin ad icin:

public String getIsYerAdi()

public void setIsYerAdi(String isYerAdi)

IsIlan – *Şirketlerin ilanlarını heap öncelik kuyruk şeklinde tutar.*

Metotlar:

Bilgileri almak ve atamak için getter ve setter metotlar örneğin iş tanıtımı icin:

public void setIsTanimi(String isTanimi)
public String getIsTanimi()

Ilanı gösterir.

public void displayIlan()

Sirket – *Şirket için bir sınıftır. Şirket bilgiyi tutan işyer bilgi ve ilanların tutan işilan nesnelerin referansını tutar.*

Metotlar:

Getter,Setter ve constructor içermekte.

<u>Dosya Package</u> – *içinde olan sınıdflar* : **Application, DosyaIsilem**

DosyaIsilem sınıfı – *dosya işlemlerini gerçekleştirir. "eleman.txt" ve "sirket.txt" dosyaları işletir.*

public void openFile(IsciDugum node,String dosyaAdi)
dosya adindan file olusturur ve recursiveYaz metodu cagirir

private void recursiveYaz(IsciDugum node,BufferedWriter out)
agaci recursive dolasarak dosyaya yazar

public void okuEleman(String dosyaAdi,IsciBinaryTree node)
isciye iliskin dosyayi acir ve dosyadan okuyup agaca ekler

public Hashtable<String, isyerBilgi> okuSirket(String dosyaAdi)
sirkete iliskin dosyayi acip okur ve hashtable-e ekler

public void yazSirket(Hashtable<String, isyerBilgi> table, String dosyaAdi)
sirket bilgiler hashtableden cekip dosyaya uygun sekilde yazar

Application sınıfı – *programı yürütür.*

Metodlar:

public static void main(String[] args)
dosyalari acip agac ve hashtabe olusturacak ve programin calismasi icin menu metodu cagirir. Program bittiginde hashtable ve agaci dosyaya yazar.

public static void menu(IsciBinaryTree tree,Hashtable<String, isyerBilgi> table)
kullanic secimine gore ilgili metodlari cagirip kullanici istegini gerceklestirir.

private static void secim1(IsciBinaryTree tree,Hashtable<String, isyerBilgi> birHashtable)
iscilerin kullanicagi bolum

private static void sistemeKayit(IsciBinaryTree currenTree)
iscilerin bilgilerini alip sisteme kaydeder

private static void deneyimEkle(Vector<IsDeneyimi> denetim)
sistemeKayit metodun yardimci metodu

private static void egitimDurumEkle(Vector<EgitimDurum> egitimDrum)
sistemeKayit metodun yardimci metodu

private static void sistemdeGunleme(IsciBinaryTree tree)
iscinin istedigi bilgiyi gunceller

private static void birIseBasvurma(Hashtable<String, isyerBilgi> birHashtable,IsciBinaryTree currentTree)
iscinin ilan verilmis sirketin bir ise basvurmayi saglar

private static void secim2(Hashtable<String, isyerBilgi> table)
sirketlerin kullanicagi bolumu

private static void sistemeIsyerekle(Hashtable<String, isyerBilgi> table)
sirketlerin bilgilerini alip sisteme kaydeder

private static void sistemdeGunlemeIsyer(Hashtable<String, isyerBilgi> table)
sirketin istedigi bilgiyi gunler

private static void birIlanVerme(Hashtable<String, isyerBilgi> currentHashtable)
sirketin verdigi bir ilani alıp heape ekler

private static void iseVuranlariListeleme(Hashtable<String, isyerBilgi> currentHastable)
ilgili sirketin ilana basvuran iscilerin bilgilerin listeleme

private static IsciDugum enUygunAdayiIseAlma(Hashtable<String, isyerBilgi> currentHashtable)
sirketin ilgili ilana en uygun kisiyi heapten ceker(en yuksek puani olan)

private static isIlan isIlaniniCekme(Hashtable<String, isyerBilgi> currentHashtable)
ilani ilgili vectordan siler

private static void secim3(IsciBinaryTree tree)
agac islemler gerceklestirir

Dosya Package– *içinde olan sınıflar*: **Node, Heap**

Node sinifi- *basvuranlarin ise uygunlugunu gosteren puani ve isciDugumun referansini tutar*

Heap sinifi- *node tipinde vector tutar ve agaca ekleme silme islemi*

Metodlar:

public int insert(double key)
ise uygunlugu yuksek olan rootte tutulur ve duzey arttikca uygunlugu azalacak sekilde heape ekler

public void trickleUp(int index)
eklerken buyukten kucuge(agac) sekilde ekler

public Node remove()
rootu siler

public void trickleDown(int index)
root silindikten sonra rootun yerine gececek elemani bulur ve heape duzenler

1.4-Veri Yapıları

Programda **ağaç , heap, vector ve hashtabe** yapıları kullanılmıştır.

TREE:

|
|IsciDugum
| |
| |**String** ad , adresi, telefon, e-posta, uyruğu, eposta
| |**String** doğum yeri, int doğum tarihi, medeni durum, String yabancı dil
| | **String** ilgi alanları, referans olan kişiler

| | `Vector<IsDeneyimi> isDeneyim`
| || | **String** ad, adres, pozisyon, gorev

| | `Vector<EgitimDurum> birEgitim`
| || | **String** okulAd, bolumu
| || |**int** baslangic,ve bitis tarih
| || |**double** notOrtalamasi

| | `IsciDugum leftChild,rightChild`
| | **int** height

Hashtable:

|
|isYerbilgi
| | **String** isyerAd, adresi **int** telefon,faks
| |**Vector<isIlan>** isIlanVector
| || |**String** isTanimi, elemanOzellikleri
| || |**Heap** ilanHeap
| || || |**Heap**
| || || || |**Vector<Node>** heap
| || || || |**int** currentSize

| || || || |**Vector<Node>**
| || || || || |**double** iseUygunlugu
| || || || || |**isciDugum** birIlaniIliskinBasvurularinReferans

Heap hashtabel icinde girilmistir

1.5-Dosya Özellikleri

ObjectInputStream ve ObjectOutputStream nesneleri ile nesneler direk bellekten alınıp diske yazılır. Böylece metin metin eklemeye gerek kalmaz. Aynı durum yükleme için de geçerlidir.

Onun için tüm bilgiler tek dosyada tutulabilir. Yani içindeki tüm bilgilerle birlikte nesne diske yazılmış olur. Dosyanın okunması ve yazılması programın başında ve sonunda yapılır. Programın işleyişi sırasında dosya kulla-nılmaz her şey bellekte gerçekleşir.

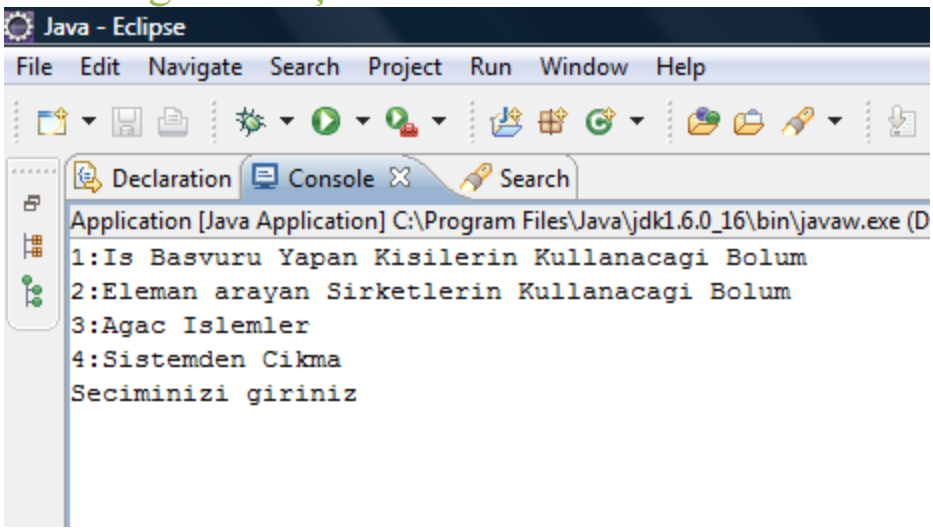
En sonunda eleman.txt ve sirket.txt dosyalarina yazılır. Söylediğimiz gibi dosya programın başında okunur, sonunda yazılır ve bu işlemleri Maindeki okurken okuEleman ,okuSirket ve dosyaya yazarken openFile, yazSirket metotları yürütür.

1.6-Çalışma Süreleri

Projenin tamamlanması, karar verme süreci dahil olmak üzere toplamda yaklaşık 10 gun almıştır. Bu ortalama geçen süre içerisinde her bir çalışan kendi esnekliğine bağlı olarak günde en az 4 saatlik çalışma yapmıştır.

2. Kullanıcı Kataloğu:

2.1-Programın İşletimi



Program *sirket* ve ise *girmek isteyen elemanlara* göre iki ayrı bölüm içermektedir.Sirket taraf ve elemanlarin ortak taraf is sisteme yeni bilgi-ler ekleyebilir, sistemdeki bilgilerde değişiklik yapabilir ve sistemden bilgileri silebilir sitemdeki bilgilerini goruntubilir. Bunu disinda sirket taraf yeni bir is ilan verebilir,ilanlar basvuranlardan ise alabilri , hic eleman almadan ilani silerbilirler elemnan taraf ise sirketlere basvurabilirler.Sirket ve elemanlar menüler içerisinde dolanarak işlemlerini ger-çekleştirip programdan çıkar.

2.2-Kullanici Kilavuzu

Program baslatildiginda bir 4 siktan olusan menu sunulur.

1 sik: Is arayanlar tarafindan kullanilacagi bolum ve icinde 5 siktan olusur.

1 sik: Yeni kayit.Eger kayitli degilsen buraya girip yeni kaydini yapabilirsin.Burad sirkete ise alinmak icin gerekli bilgileri ister

2 sik: Sisteme kayitli ise bilgilerin girip gunleme yapma.

3 sik: Sisteme kayitli ise bilgilerin slime.Sadece ismini istener.

4 sik: Sirketlerin verilmis liana basvurma. Hangi sirkete basvuracaginin sorur.Daha sonrasindan sizin kayitli olup olmadigini control etmek icin isminizi ister kayitli ise ilgili sirketin ilanlar karsiniza cikar. Numarsini girip basvurursunuz.

5 sik: Ust ana menuya donus.

2 sik: Sirket tarafindan kullanilacagi bolum ve icinde 7 siktan olusur.

1 sik: Sisteme sirketler kayit yapmak icin.

2 sik: Ilgili sirket kayitli ise gunleme yapilir

3 sik: Sirket kayitli olup olmadigin sirket adini isteyip kontol ettikten sonar kayitli ise yeni ilan bilgilerini alinir

4 sik: Ilgili sirket ilanlarina basvuranlarin listesi

5 sik: Ilgili sirket hangi ilandan eleman ise almak isiyors onu girer ve eleman ise alinir

6 sik: Ilgili sirket veren ilanlardan eleman ise almadan geri cekme

7 sik: Ana menuye donme

3 sik: Agac islemler 5 siktan olusur.

1 sik: Adından kişi arama, tüm bilgilerini listeleme (başvurduğu işlerle birlikte).

2 sik: Not ortalamalarından en az birisi, 90'ın üzerinde olan kişilerin adlarının listelenmesi.

3 sik: İngilizce bilen kişilerin adlarının listelenmesi.

4 sik: İkili arama ağacındaki tüm kişilerin adlarını düzeyleri ile birlikte Listeleme (Inorder, preorder, postorder). Ağacın derinliğini ve eleman sayısını yazdırma.

5 sik: Ana menuye donme

4 sik: Sitemden cikis.

2.3-Programin Kisitlamalari

Veri girislerini dogru sekilde girmezse programin ortasinda hata verip cikabilir(bazi yerlerde hata kontrolu yapilmistir).