Rezervasyon Sınıflandırma Sistemi

\*Ceren Genç, \*Berkehan Öztürk

\*Bilgisayar Mühendisliği Bölümü, Kocaeli Üniversitesi

Kocaeli, Türkiye

[cerengenc17@gmail.com](mailto:cerengenc17@gmail.com)

[berkehanozturk98@gmail.com](mailto:berkehanozturk98@gmail.com)

**Anahtar Kelimeler – “rezervasyon.txt”, “Kullanıcı”, “Kategori”, “Alt kategori”, “Kategori yolu”**

Özet – Programımız “rezervasyon.txt “ dosyası içinden kayıtlı rezervasyon bilgilerini okur ve kategorileri gereken kategori düğümleri altına bağlayarak ağaç yapısı oluşturur.

I. GİRİŞ

Programımız “rezervasyon.txt” dosyası içinden bilgilerin okunması ve kullanıcı düğümlerinin oluşturulması ile başlar. Kullanıcı düğümleri oluşturulurken aynı kategoride yapılmış rezervasyonlar bağlı liste yapısı içinde tutulur. Kullanıcı düğümleri içinde gezilerek rezervasyon yapılmış tüm kategoriler tekrarsız şekilde gereken düğüm altına yerleştirilir ve ağaç yapısı oluşturulmuş olur. Kategori düğümlerinin altına bu kategoride rezervasyon yapmış kullanıcılar Heap ağaç yapısı kullanılarak eklenir. Heap ağacına eklenen her kullanıcının bu kategoride yapmış olduğu rezervasyonların tutulduğu bağlı liste yapısı düğüm yapısına bağlanır. Gerekli yapılar oluşturulduktan sonra kullanıcıya hangi işlemi yapmak istediği sorulur. Kullanıcının seçtiği işleme göre işlemin gerçekleştirildiği fonksiyona gönderilir. Kategori bulma işleminde ağaç yapısı üstünde gezilerek gerekli kategori bulunur. Parent düğümü ve düğüme bağlı çocuk düğümler kullanıcıya gösterilir. Aranan kategori ağaç yapısı içinde bulunmuyorsa işlem sonlandırılır. Kategori düğümü bulunmuş ise bu kategorinin altına yeni bir kategori düğümü eklemek isteyip istemediği kullanıcıya sorulur. Yeni kategori ekleme işleminde ise ana kategori ekleme işlemi yapılır. Eklenmek istenen kategori ağaç yapısı içinde mevcutsa işlem sonlandırılır. Kategori silme işleminde silmek istenilen kategorinin Heap ağaç yapısının bağlantısı kesilir. Kategoriye bağlı alt düğümler mevcutsa bir üst kategoriye bağlanır. Kullanıcı ekleme işleminde öncelikle kullanıcının hangi kategoriye eklenmek istediği sorulur. Eklenmek istenilen kullanıcı mevcut ise işlem sonlandırılır. Yoksa yeni kullanıcı düğümü oluşturularak ilgili kategorinin Heap ağacına eklenir. Kullanıcıya göre kategori listeleme işleminde kullanıcının bağlı liste yapılarının isimleri yazdırılarak kullanıcının rezervasyon yapmış olduğu kategoriler tekrarsız bir şekilde kullanıcıya gösterilir. Kategoriye göre kullanıcı listeleme işleminde ağaç yapısı içinde gezilerek istenilen kategorinin Heap ağacındaki kullanıcı isimleri yazılır. Rezervasyon yerine göre kullanıcı listeleme işleminde, kullanıcı düğümleri içinde gezilerek aynı yerde rezervasyon yapmış olan kullanıcılar ArrayList yapısı içinde tekrarsız bir şekilde tutulur ve kullanıcıya gösterilir. Kullanıcıya göre rezervasyon listeleme işleminde kullanıcın tüm rezervasyon bilgileri gezilir. Aynı yerde birden fazla rezervasyon yapılmış ise son rezervasyon tarihi ile kullanıcıya gösterilir.

II. TEMEL BİLGİLER

Programımız Java dilinde yazılmıştır. Geliştirme ortamı olarak “NetBeans 8.2” kullanılmıştır. Windows 10 ve Windows 7 işletim sistemlerinde sorunsuz olarak çalıştırılmıştır.

III. MİMARİ

Gerekli yapıların oluşturulabilmesi için sınıf yapıları kullanılmıştır. Tree sınıfı içinde ağaç yapısı oluşturulmuştur. Bu sınıfın içinde tree tipinde child ArrayList’i, tree tipinde parent, String tipinde name, node tipinde kullanıcı, integer tipinde derinlik tutulmuştur. Root ana düğümünün çocuk düğümlerine ana kategoriler bağlanmıştır. Kategorinin alt düğümleri mevcut ise gerekli işlemler yapılarak o kategori düğümünün altına bağlanır. Kullanıcı sınıfında kullanıcı düğümleri oluşturulmuştur. Bu sınıf içinde bağlı liste tipinde ArrayList ve String tipinde userID tutulur. Her yeni kategorideki rezervasyonlar için yeni bir bağlı liste yapısı oluşturulup ArrayList’e eklenir. Aynı kategori için yapılmış rezervasyonlar aynı bağlı liste yapısına bağlanır. Dugum sınıfı içinde rezervasyonların bilgileri tutulur. Hepsi string tipinde olmak üzere placeID, city, lat, lon, dateTime ve dugum tipinde önceki ve sonraki bilgileri tutulur. linkedList sınıfında bağlı liste yapısı oluşturulur. Bağlı listenin elemanları dugum tipindedir. Bu sınıf içinde tree tipinde parent, dugum tipinde first ve end, String tipinde userID, kategoriYolu ve name, int tipinde boyut değişkenleri tutulur. Node sınıfı içinde heap ağaç yapısı oluşturulur. Bu sınıf içerisinde tree tipinde kategori, String userID, integer tipinde key, number, sayaç ve kullanıcıSayisi, node tipinde left, right, parent, linkedList tipinde rezervasyon bilgileri tutulur.

IV.KULLANILAN FONKSİYONLAR

1. *Boolean varMi():* Fonksiyona yollanan ArrayList yapısı içinde kategori adının bulunup bulunmadığını kontrol eder.
2. *Void addUsers():* Ana kategorilerin heap ağacını oluşturur.
3. *Void addUsers2():* Birinci alt kategorilerin heap ağacını oluşturur.
4. *Void addUsers3():* İkinci alt kategorilerin heap ağacını oluşturur.
5. *Void kategoribulma():* Kategori bulma işlemini gerçekleştirir. Bulunan kategorinin çocukları gösterilir. Bu kategorinin altına yeni kategori eklemek isteyip istemediği sorulur. İstenirse ekleme işlemi gerçekleştirilir.
6. *Kullanicikategoriliste():* İstenilen kullanıcının rezervasyon yaptığı kategoriler listelenir.
7. *Void kategoriEkle():* Ana kategori ekleme işlemi gerçekleştirilir. Eklenmek istenilen kategorinin var olup olmadığı kontrol edilir. Yoksa ekleme işlemi gerçekleştirilir.
8. *Void kategoriSil():* Silinmek istenilen kategorinin çocukları bir üst düğüme bağlanır. Kullanıcı heap ağacı bağlantısı kopartılır.
9. *Void kategoriKullaniciListele():*İstenilen kategoride rezervasyon yaptırmış kullanıcılar yazdırılır.
10. *Void rezervasyonYeriListele():* Aynı yerde rezervasyon yaptırmış kullanıcılar listelenir.
11. *Void kullaniciRezervasyonListele():* Kullanıcının farklı yerlerde yaptırmış olduğu rezervasyonlar listelenir. Aynı yerde birden fazla rezervasyon yaptırılmış ise son rezervasyon tarihi ile listelenir.
12. *Void kullaniciEkle():* Hangi kategori altına kullanıcı eklenmek istediği kullanıcıya sorulur. Kullanıcının var olup olmadığı kontrol edilir. Yoksa ilgili kategori altına ekleme işlemi gerçekleştirilir.
13. *Void addChild():* Ana kategoriler ağaç yapısına eklenir.
14. *Void addChild1():* Birinci alt kategoriler ağaç yapısına eklenir
15. *Void addChild2():* İkinci alt kategoriler ağaç yapısı altına eklenir.
16. *addNew():* Yeni kategori ekleme işlemi yapılırken ağaç yapısına kategoriyi ekler.
17. *Void addChild3():* İkinci alt kategorinin altına yeni kategori ekleme işlemi gerçekleştirilir.
18. *Void kullaniciNode():* Yeni kullanıcı node’u oluşturur.
19. *Void addKategori():* Kullanıcının bağlı liste ArrayList yapısında kategoriye ait bağlı liste yapısı yoksa yeni bağlı liste oluşturur. Varsa ilgili bağlı listeye yeni düğüm ekler.
20. *Void addNode():* Var olan bağlı listeye yeni dugum ekler.
21. *Void addList():* İlgili bağlı liste yapısına yeni düğüm yapısı ekler.
22. *Void addNode():* Heap ağaç yapısına kullanıcı ekler.
23. *Void inOrder():* Heap ağaç yapısına eklemenin soldan sağa eklemeyi gerçekleştirir.

V. SONUÇ

İstenilen yapılar oluşturulmuştur. İsterler içinde kullanıcı silme işlemi gerçekleştirilememiştir. Heap ağaç yapısı oluşturulmuş fakat Max-Heap yapısına dönüştürülememiştir.

VI. KABA KOD

1-Başla

2-File tipinde rezervasyon.txt açılır

3-Rezervasyon.txt sonuna kadar okunur

4-İlk satır kullanici node'una eklenir

5-Kullanici node'unda gelen satır , lere göre ayrılır

6-Ayrılan kelimeler istenilen yerlere atanır

7-Tree tipinde root dizinimiz oluşturulur

8-Roota cocuk olarak ana kategoriler baglanir

9-Eklenecek kategoriler agaçta dolaşılarak kontrol edilir yok ise eklenir var ise boş geçilir

10-Kullanicilarin kategorileri :'ya göre ayrılıp kategori yoluna eklenir

12-Kullanicilar Heap agacına eklenir

13-Yapılacak işlemler sorulur

14-Seçilen işleme göre ilgili fonksiyona gönderilir

15-Kategori bulma işlemi için Kategoribulma fonksiyonuna

16-Yeni kategori eklemek için kategoriEkle fonksiyonuna

17-Kategori silmek için kategoriSil fonksiyonuna

18-Kullanıcıya göre kategori listelemek için kullanicikategoriliste fonksiyonuna

19-Kategoriye gore kullanici listelemek için kategoriKullaniciListele fonksiyonuna

20-Rezervasyon yerine kullanici listelemek için rezervasyonYeriniListele fonksiyonuna

21- Kullaniciya gore rezervasyon listelemek için kullanicirezervasyonliste fonksiyonuna

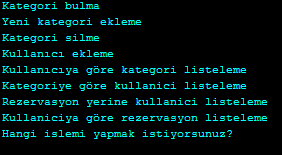
22-Kullanici eklemek için kullaniciEkle fonksiyonuna gönderilir

VII.KAYNAKÇA

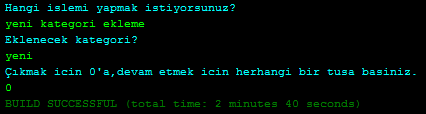
<https://stackoverflow.com/questions/23673180/create-generic-tree-with-more-than-two-child-each-may-have-unique-proprties>

<https://gelecegiyazanlar.turkcell.com.tr/konu/android/egitim/android-101/javada-dosya-islemleri>

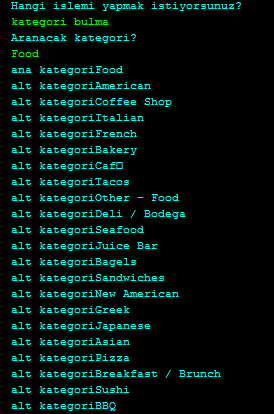
<https://sanalkurs.net/java-ile-agac-yapisi-9424.html>

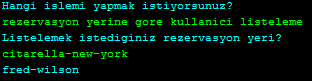
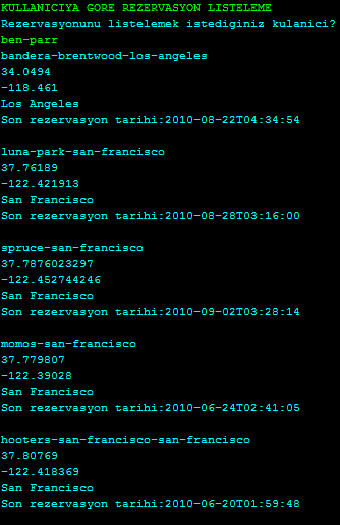
VIII.EKRAN GÖRÜNTÜLERİ

1.SEÇİM EKRANI

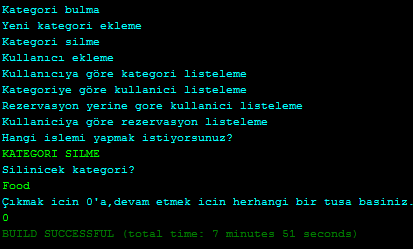
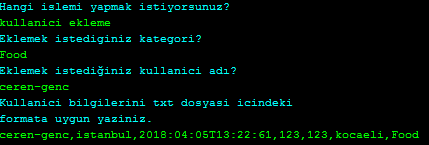
2. KULLANICIYA GÖRE KATEGORİ LİSTELEME

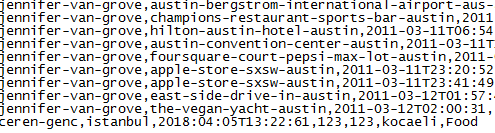
3. YENİ KATEGORİ EKLEME

4. KATEGORİ BULMA

5.REZERVASYON YERİNE GÖRE KULLANICI LİSTELEME

6. KULLANICIYA GÖRE REZERVASYON LİSTELEME

7. KATEGORİ SİLME

8. KULLANICI EKLEME

9. REZERVASYON.TXT DOSYASININ GÜNCELLENMESİ