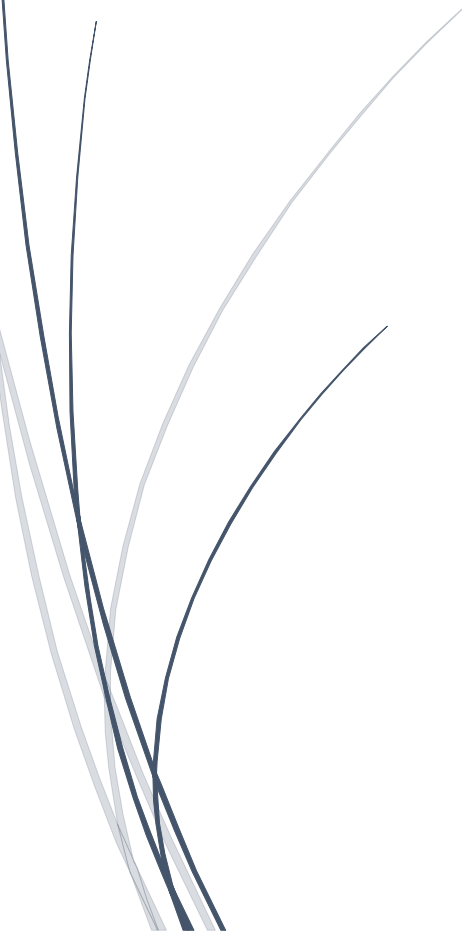




02.12.2016

CSE 443

Ödev2 Raporu



Merve Şahin
GEBZE TEKNİK ÜNİVERSİTESİ

PART-1)

Ödevin ilk partında e-posta adreslerini içeren bir e-posta adres defteri implement edilecektir.

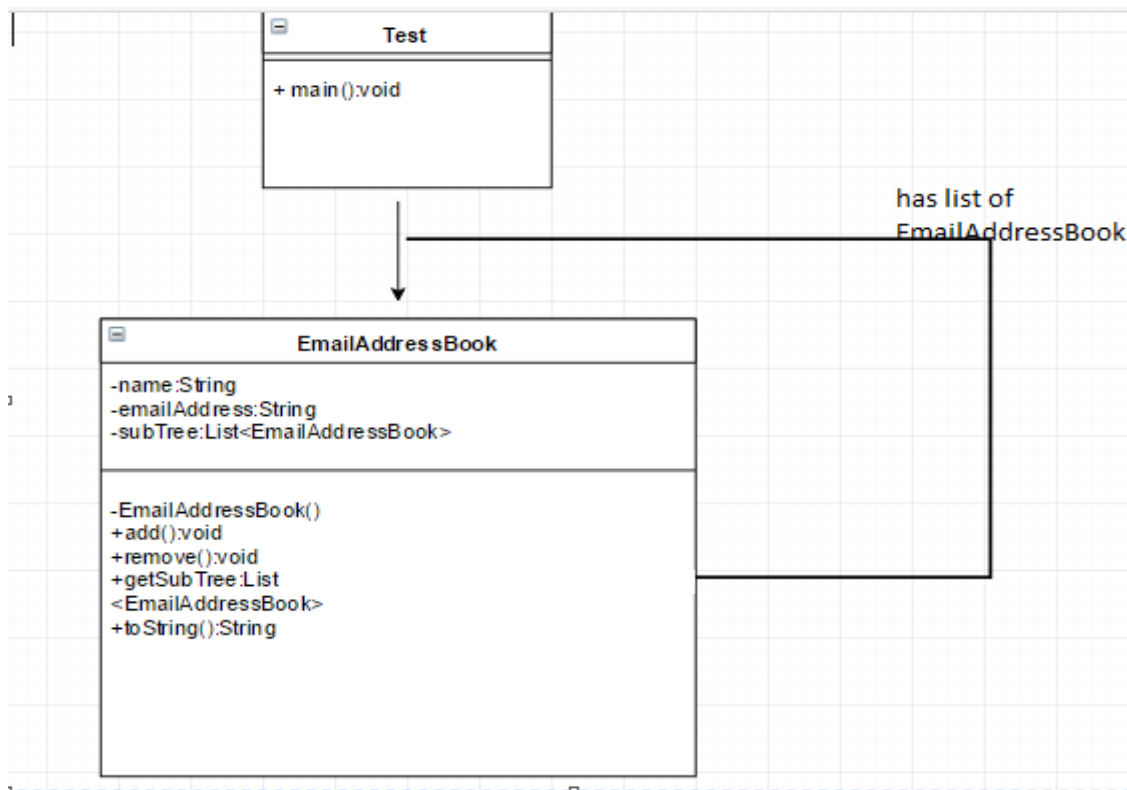
E-posta adresleri kişiye ya da gruba ait olabilir. Her kişi için isim ve mail adresi olması gerekmektedir. Grup mail adresleri kişi mail adreslerini barındırır. Aynı zamanda tekrar bir grup adresi de barındırabilmektedir.

Programda yapılması istenen mail adreslerinin ekrana print etmesidir. Grup mail adresleri kişi mail adreslerini de bünyesinde bulundurabildiği için bu yapı tree yapısıdır.

Tasarım da composite – birleşik tasarım örünüdür kullanılmaktadır.

Kendi içlerinde birbirlerinden farklı olan nesnenin sanki bir bütün nesneymiş gibi kullanılmasını composit – bileşik tasarım örüntüsü sağlar. Nesneleri bir tree yapısında birleştirip tasarımın genelinde parça bütün ilişkisini düzenleyip şekillendirmektedir.

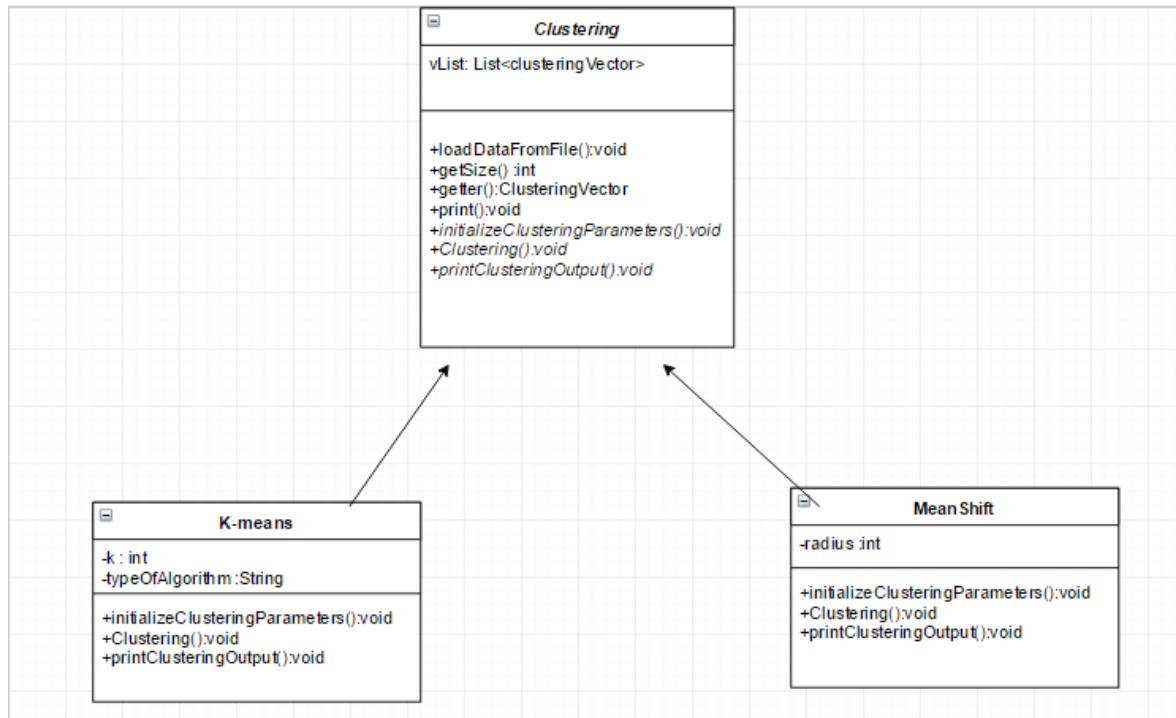
Tasarımın Sınıf diyagramı aşağıda şekilde gösterilmiştir.



(Part-2)

Template tasarım deseni ;behavioral grubuna ait bir algoritmanın adımlarının sınıfta tanımlanarak farklı adımlarının concrete sınıflarında overwrite edilip çalıştırılmasını sağlar.

Template tasarım deseninde bir operasyonu gerçekleştirmek için gerekli olan adımlar ve bu adımları çalıştıracak metotlar abstract bir sınıfta tanımlanır.Bu abstract sınıfı uygulayan gerçek sınıflar gerekli adımları overwrite eder ve abstract sınıfta bu adımları kullanan metotlar çağrılarak işlem gerçekleştirilir.Böylece kod tekrarı yapılmasının önüne geçilmiş olunur.Template tasarım deseninin part2 için uml şeması aşağıdaki gibidir.



Ödevde data dosyası içerisinde x,y,z olmak üzere 3 boyutu olan noktalar tanımlıdır.Bu data 2 şekilde kümelere ayrılacaktır.Bunun için kullanılacak olan 2 çeşit kümeleme algoritması vardır.K-means ve MeanShift kümeleme yöntemleri. Temelde ikisinde kümeleme yapar.Ama birbirinden farklıdır.Mesela K-means algoritması kaç kümeye ayıracağını kullanıcıya sorar.MeanShift ise (parametrik olmayan kümeleme yöntemi olması dolayısıyla) sadece bir r Yarıçap bilgisi ile işlem görür.

Burada bizi template tasarım örüntüsü kullanmaya iten şey; Bu ikisinde kümeleme yapmak için kullanacağı ortak operasyonlar iki kümeleme yöntemi de kümeleme yapmak için file dan data yı okumak zorundadır mesela. Bu nedenle Clustering adında abstract bir sınıf yaptım. Ve ortak olan metodu ise orada implement ettim.(Uml dende anlaşılacağı üzere Clustering italik olarak yazılmıştır. Buradan soyut bir sınıf olduğu anlaşılmaktadır.)Clustering sınıfı concrete

sınıflarda tanımlanacak olan metodları abstract tutmuştur. Ve burada implement edilmek zorunda değildir. Aynı zamanda 3 boyutlu datayı tutabilmek için ClusteringVector adında bir sınıf oluşturdum ve bu sınıf çok basit bir şekilde private değişken olarak x,y,z bilgilerini tutmakta ve bunların setter ve getter metodlarını bulunmaktadır.

Clustering soyut sınıfım bu vektör classının elemanlarını bir arraylist olarak tutmaktadır.

Bu soyut sınıfı extend eden K-means ve meanShift concrete sınıfları ise object oriented düzeni bozmamak adına bu datanın size bilgisine ve herhangi bir i indexindeki vektör verisine

Ödev formatında yazmayan aşağıda tanımları verilen metodlar sayesinde ulaşmaktadır.

```
public clusteringVector getter(int i){
    return vList.get(i);
}
public int getSize(){
    return vList.size();
}
```

K-means algoritması kullanıcıdan küme sayısı ve distance hesaplarken kullanacağı algoritma tipini kullanıcıdan aldıktan sonra alınan küme sayısı kadar tane random küme merkezi oluşturur. Daha sonra ise dataset içerisinde yer alan tüm verilerle random merkezleri arasında uzaklık hesaplar. Datalar hangi merkeze daha yakın ise artık ona ait bir etiket alırlar. Aldıkları etikete göre de kendi içlerinde bir ortalama hesabına giderek random olarak seçilen nokta verisini güncellerler. Bu işleme seçilen merkez verileri değişmeyene ve ya belli bir epsilon değerinin altına varılana kadar devam edilir. En sonunda girilen grup sayısı kadar küme merkezi outputunu verir.

```
public class Kmeans extends Clustering{
    int k ; //grup sayisi
    String typeOfAlgorithm;
    public Kmeans () {
        k = 2; //default olarak 2 gruba ayırsın.
        typeOfAlgorithm = "Euclidian";
    }

    public void initializeClusteringParameters () {
```

Yukarıda Kmeans algoritması için yazılan sınıf gösterilmiştir. K means algoritması için uzaklık hesaplama algoritmaları şu şekildedir.

Difference between Euclidean and Manhattan

edureka!

From this image we can say that, The Euclidean distance measure gives 5.65 as the distance between (2, 2) and (6, 6) whereas the Manhattan distance is 8.0

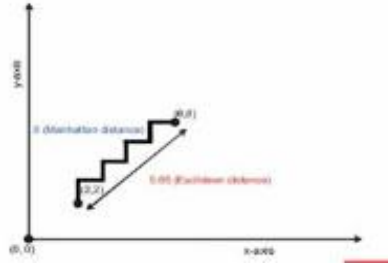
Mathematically, Euclidean distance between two n-dimensional vectors

(a_1, a_2, \dots, a_n) and (b_1, b_2, \dots, b_n) is:

$$d = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + \dots + (a_n - b_n)^2}$$

Manhattan distance between two n-dimensional vectors

$$d = |a_1 - b_1| + |a_2 - b_2| + \dots + |a_n - b_n|$$



Euclidean ve Manhattan olmak üzere 2 yöntem vardır. Yöntemlerin matematiksel kısmı yukarıda gösterilmiştir ve şu şekile implement edilmiştir.

```
//distance = Noktaların farklarının karelerinin toplamının karekökü Euclidean ile
private double CalculateDistanceForEachCenter(clusteringVector dot1,clusteringVector dot2){
    int distanceX = dot1.getX()- dot2.getX();
    int distanceY = dot1.getY()- dot2.getY();
    int distanceZ = dot1.getZ()- dot2.getZ();
    return Math.sqrt(Math.pow(distanceX,2)+ Math.pow(distanceY,2) + Math.pow(distanceZ,2));
}

//distance = |x1-x2| + |y1-y2| + |z1-z2| with Manhattan
private double CalculateDistanceForEachCenterWithManhattan(clusteringVector dot1,clusteringVector dot2){
    int distanceX = dot1.getX()- dot2.getX();
    int distanceY = dot1.getY()- dot2.getY();
    int distanceZ = dot1.getZ()- dot2.getZ();
    return (Math.abs(distanceX)+ Math.abs(distanceY) + Math.abs(distanceZ));
}
```

MeanShift algoritması ie parametrik olmayan bir algoritmadır.Kaç tane küme olduğunu kullanıcıdan istemez.Sadece bir yarıçap bilgisine ihtiyacı vardır.Hangi nokta ile başladığının önemi yok. Bir nokta ile başlayıp etrafına bir daire çiziyor."r" yarıçaplı bu daire içerisinde kalan noktaların bilgilerini toplayıp nokta sayısına bölüp ortalama bulur. Sonra o nokta için yeni bir merkez bulur.Merkez bilgisi sürekli güncellenicek ve bir noktadan sonra duracaktır.

MeanShift concrete sınıfı abstact olan soyut sınıf metodları implement eder ve implement edilen metodları ise miras alarak kullanabilir.

Clustering sınıfı abstract olması nedeni ile objesi yapılamaz ama referansı tabiki de yapılabilir.O yüzde main de ;

```
Clustering [] rArray ; //abstract classın objesi yapılamaz referansı yapılabilir.

rArray = new Clustering[2];
Kmeans obje1 = new Kmeans();
rArray[0] = obje1;

try{
    obje1.loadDataFromFile();
}

catch(Exception e){
    System.out.println("DataFromFile exception !!!!!");
}
```

Bu şekilde yazıldı.

