

GIT Department of Computer Engineering

CSE443 - Object Oriented Analysis and Design

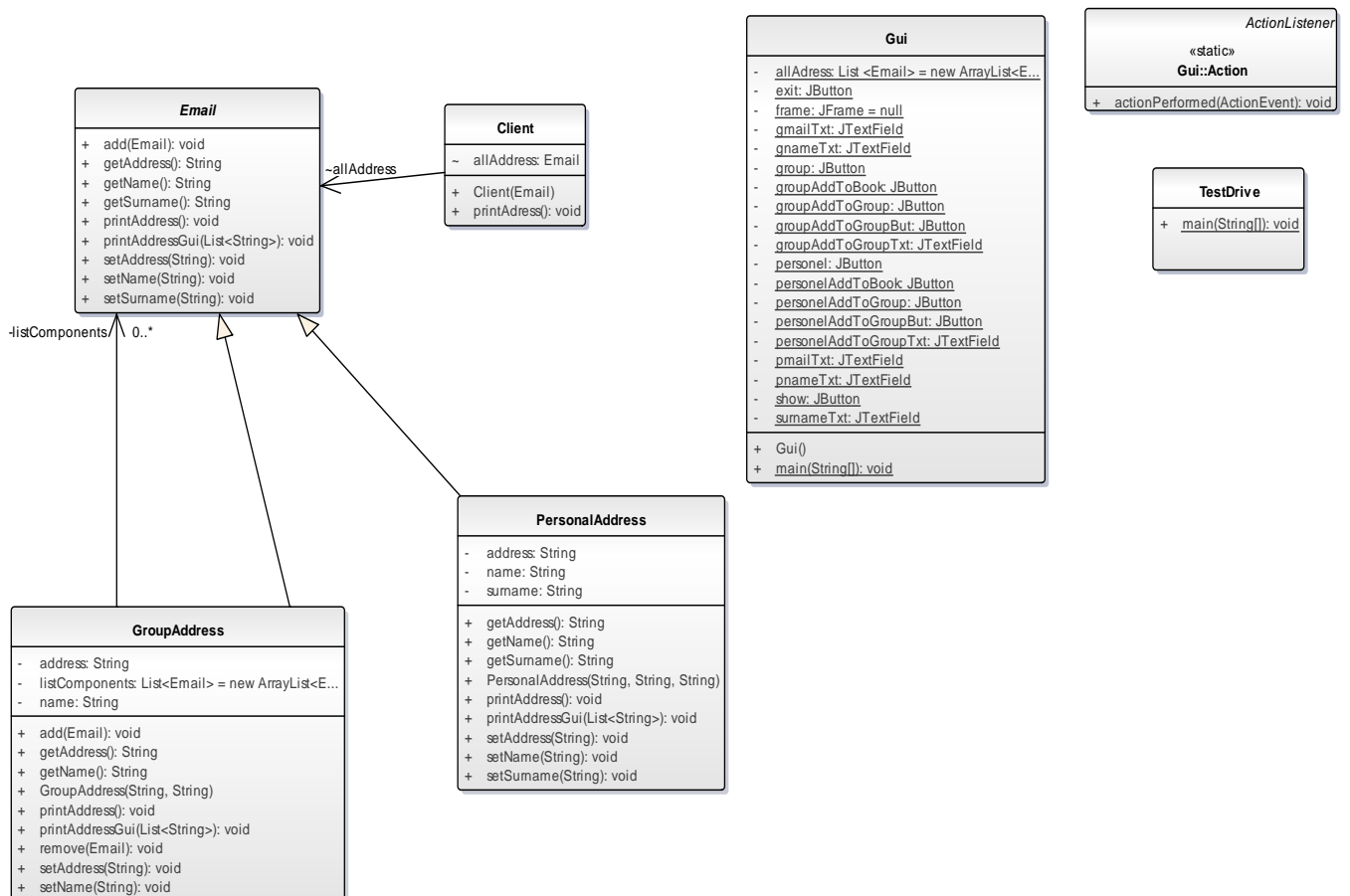
Fall 2016

HW2

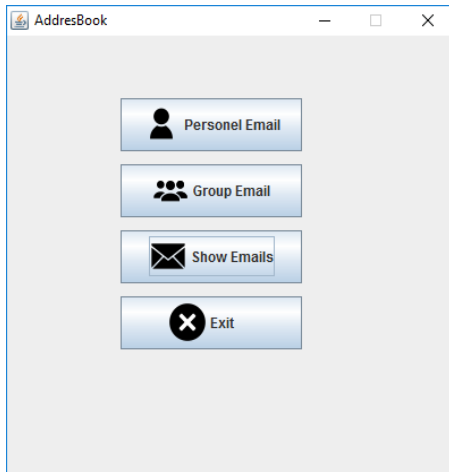
Onur SEZER
121044074

Q1)

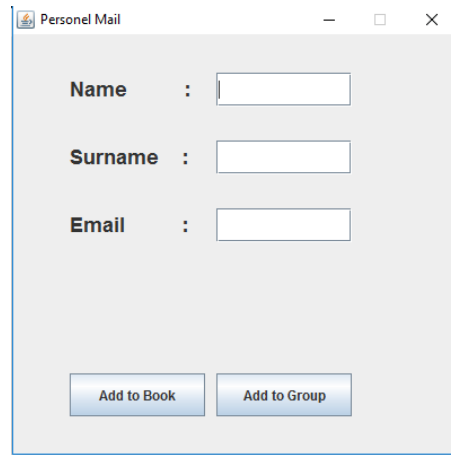
- İlk bölüm için çıkan UML diyagramı aşağıdaki gibidir.



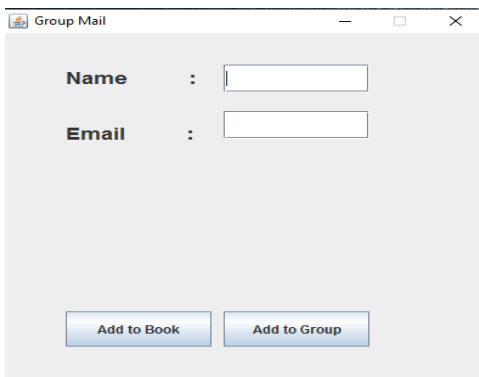
- Composite Design pattern kullanılmıştır.
- Bu pattern birbirlerinden farklı olan bir grup nesnenin sanki tek bir bütün nesneymiş gibi kullanılmasını sağlar.
- Bu patternda Bileşen(Component), Yaprak(Leaf) ve Bileşik (Composite) yapıları vardır.
- Bizim sistemimizde Email classı Component, PersonalAdress classı Leaf ve GroupAddress classı Composite görevlerindedirler.
- GroupAddress ve PersonalAdress class'ları Email abstract class'ını extend eder.
- GroupAddress class'ı personal mailleri ve group maillerini tutabilmesi için Email tipinde List tutar.
- Bu sistem için GUI yapılmıştır. Bazı işlevlerinin görünümü aşağıdaki gibidir.



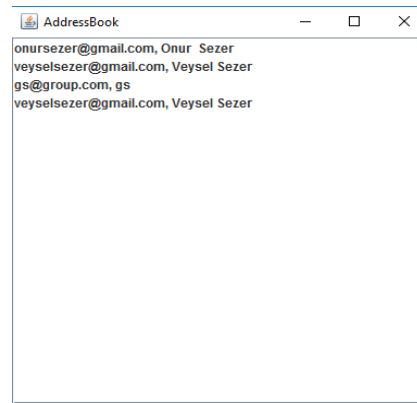
(Başlangıçta ekrana gelen frame)



(Personal Email butonuna click edildikten sonra)



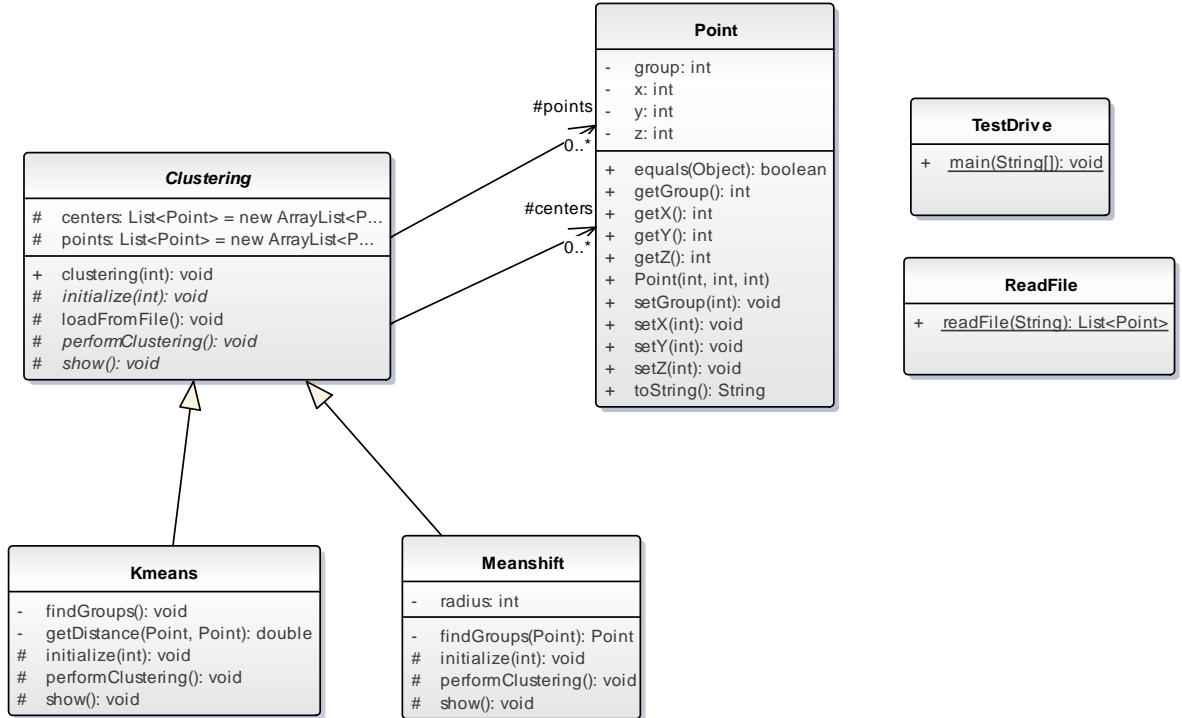
(Group Email butonuna click edildikten sonra)



(Address Book'unda olan mail'ler)

Q2)

- İkinci bölüm için çıkan UML diyagramı aşağıdaki gibidir.



- Template Method design pattern kullanılmıştır.
- Bu patternde aynı sıralı operasyonları yapan classların için kullanılır. Super classda ortak olan işlemler implement edilir değişime açık olan operasyonlar alt sınıflarda implemente edilir.
- Bizim sistemizde dosyadan okuyup bir listeye doldurma işlemini yapan `loadFromFile` methodu **Kmeans** ve **Meanshift** algoritmalarında aynı olduğu için süper class da implemt edildi.
- İki kümeleme yöntemi de çıktı olarak buldukları küme merkezlerini print eder. Örnek olarak küme sayısı 2 verirse **Kmeans** algoritmasının çıktısı:

```
For kmeans the clusters are :
Point [x=129, y=46, z=76],
Point [x=213, y=133, z=124]
```

- Meanshift** algoritmasında looptan çıkılamadı.