

CSE443 Object Oriented Analysis and Design

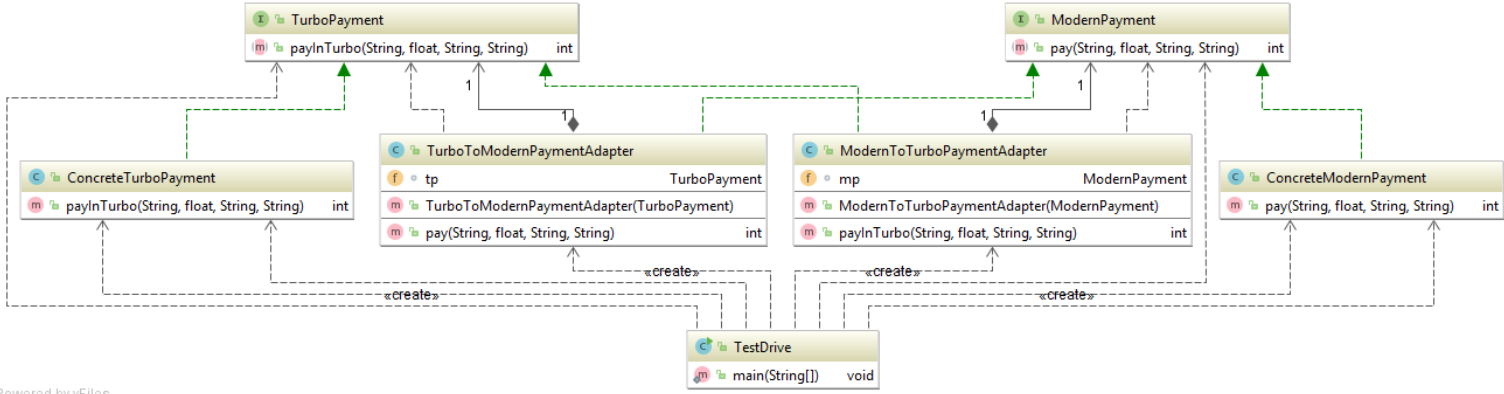
Seyit Ahmet KARACA - 141044084

HW3 Rapor

1.Soru1 - Payment	2
1.1. Payment UML Diyagram	2
1.2. Payment Design Pattern	2
1.3. Payment Sonuçlar	3
2.EmailAdressBook	3
2.1. EmailAdressBook UML Diyagramı	3
2.2. EmailAdressBook Sınıf Yapısı	4
2.3. EmailAdressBook Sonuçlar	4
3. Gokturk 3 Satellite	4
3.1. Gokturk 3 Satellite Amacı	4
3.2. Gokturk 3 Satellite UML Diyagramı	5
3.3.Gokturk 3 Satellite Sınıf Yapısı	5
3.4. Gokturk 3 Satellite Çıktıları	5
4.Transform - Template Method Design.....	5
4. 1. Transform – Template Method Design Amacı	5
4.2. Transform – Template Method Design UML Diyagramı.....	6
4.3. Transform – Template Method Design Sınıf Yapısı	6
4.4. Transform – Template Method Design Sonuçlar	7

1.Soru1 - Payment

1.1. Payment UML Diyagram



Powered by yFiles

1.2. Payment Design Pattern

Problemde adapter tasarım örüntüsü kullanıldı. Problem verilen iki interface'in birbirlerine dönüştürülmesidir. Interface'ler değiştirilmeden parametreleri anlamları aynıdır fakat isimleri farklıdır. Bunu gerçeklemek için modernPayment ve turboPayment soyut sınıfları oluşturuldu. Sonrasında adapter'leri gerçeklemek için modernAdapter'i turboPayment interface'inden , turboAdapter'i de modernPayment interface'ini implement ettirildi. Bu adapterler hedef sınıfa ait bir parametre alan kurucu fonksiyonlar yazıldı ve sınıfa eleman olarak alındı. Interface'lerden gelen pay ve payTurbo methodlarını adapterlere gelen parametreler ile doldurarak sonuçlar elde edildi.

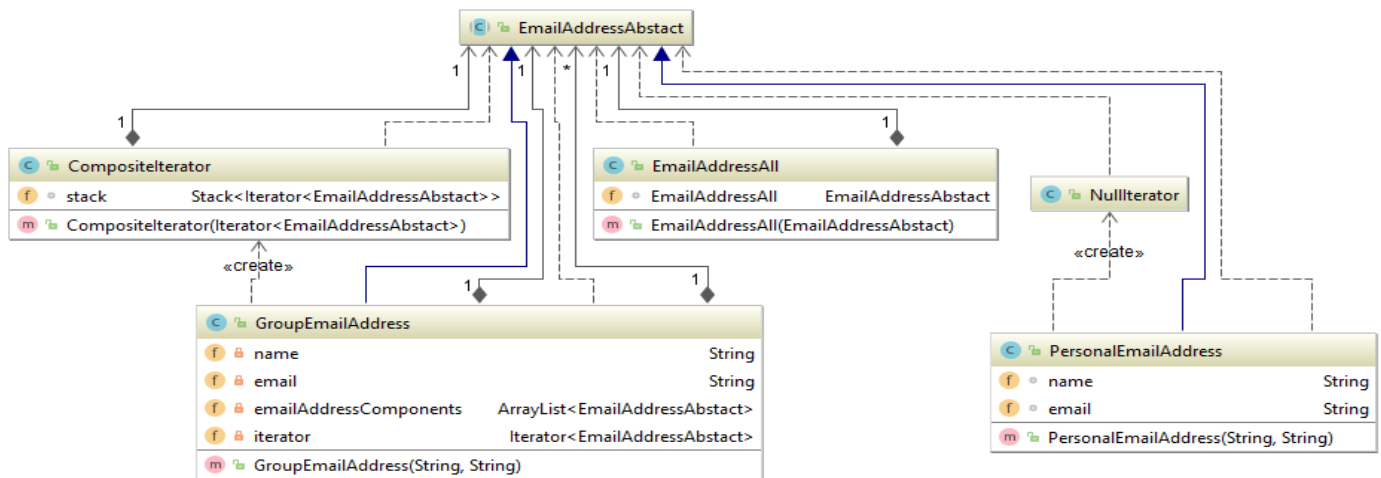
1.3. Payment Sonuçlar

```
"C:\Program Files\Java\jdk1.8.0_191\bin\java.exe" ...  
>>> Modern Payment'  
cardNo: 141044044  
destination :077  
installments :No  
-----  
>> Turbo Payment'  
turboCardNo :141044088  
destinationTurboOfCourse :97  
installmentsButInTurbo :Yes  
-----  
  
Adapter turbo -> Modern  
>> Turbo Payment'  
turboCardNo :141044088  
destinationTurboOfCourse :97  
installmentsButInTurbo :Yes  
  
Adapter Modern -> Turbo  
>>> Modern Payment'  
cardNo: 141044044  
destination :077  
installments :No
```

Çıktıda modern payment ve turbo payment'ın methodları çağrıldı ve adapter'ler ile birbirlerine çevrilerek çıktısı alındı.

2.EmailAdressBook

2.1. EmailAdressBook UML Diyagramı



2.2. EmailAdressBook Sınıf Yapısı

Projenin amacı ağaç olarak bir yapı oluşturmak ve bu yapının elemanlarına bir grup veya eleman eklenmesidir. Kişisel ve grup email adreslerini tutan sınıflar oluşturuldu. Bu iki sınıfta soyut bir email adres sınıfından türetildi. Kişisel ve grup email adreslerini tutabilen emailAdressAll isimli bir sınıf oluşturuldu. Bu sınıf sayesinde bütün emailer aynı türmüş gibi yapıda tutulabilmektedir. Her kişisel mail adresi bir gruba dahil olmak zorunda olduğu için grup email sınıfında soyut sınıftan gelen iteratör methodu implement edildi. Kişisel emailer de soyut email sınıfından türediği için burada iteratör olmaması gerektiğinden dolayı nullIterator ile bu sorun ortadan kaldırıldı.

2.3. EmailAdressBook Sonuçlar

```
ALL GROUPS

allgroups@gtu.edu.tr
-----

mailGrup-muhendis

muhendis@gtu.edu.tr
-----
MuhPersonall1-MuhPersonall1@gtu.edu.tr
MuhPersonal2-MuhPersonal2@gtu.edu.tr

mailGrup-mimar

mimar@gtu.edu.tr
-----
MimarPersonall1-MimarPersonall1@gtu.edu.tr
MimarPersonal2-MimarPersonal2@gtu.edu.tr

mailGrup-topluluk

topluluk@gtu.edu.tr
-----
topluluk-topluluk@gtu.edu.tr

mailGrup-temelBilimler

temelBilimler@gtu.edu.tr
-----
TemelBilimlerPersonall1-TemelBilimlerPersonall1@gtu.edu.tr
TemelBilimlerPersonal2-TemelBilimlerPersonal2@gtu.edu.tr
```

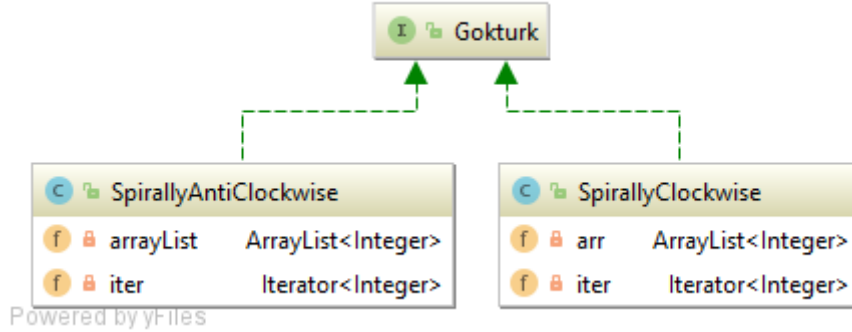
Dört tane grup oluşturuldu. Topluluk grubu mimarGruba dahil edildi. Bu gruplar kök mail adres grubuna eklendi. Görselde grupların çıktısı bulunmaktadır.

3. Gokturk 3 Satellite

3.1. Gokturk 3 Satellite Amacı

Gelen 2 boyutlu arrayi dıştan içe doğru sipral olacak şekilde iteratör ile gezmek.

3.2. Gokturk 3 Satellite UML Diyagramı



3.3. Gokturk 3 Satellite Sınıf Yapısı

İteratör interface'ini implement eden bir Gokturk interface'i oluşturuldu. İki farklı spirall gezme sınıfı yaratıldı. Bir tanesi saat yönünde diğer saat yönünün tersi yönde iki boyutlu arrayi geziyor. Bu iki sınıf ta Gokturk interface'ini implement ettiği için iteratör sınıfının Next, hasNext methodlarını yazmak zorundadır.

3.4. Gokturk 3 Satellite Çıktıları

```
Data :
1  2  3  4  5
6  7  8  9 10
11 12 13 14 15
16 17 18 19 20
21 22 23 24 25
Spirally anti clockwise
1 6 11 16 21 22 23 24 25 20 15 10 5 4 3 2 7 12 17 18 19 14 9 8 13
Spirally clockwise
1 2 3 4 5 10 15 20 25 24 23 22 21 16 11 6 7 8 9 14 19 18 17 12 13
```

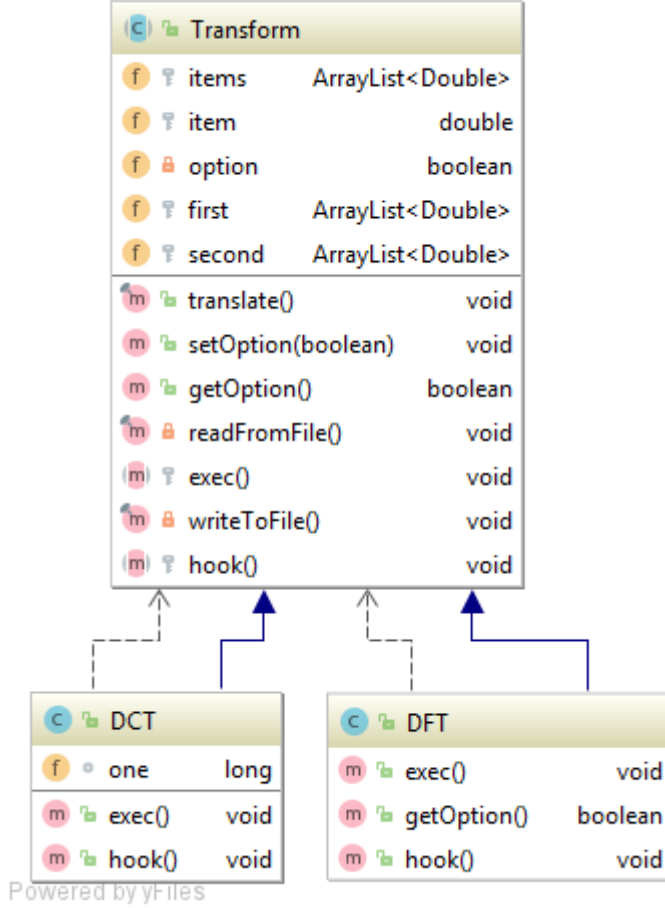
4. Transform - Template Method Design

4. 1. Transform – Template Method Design Amacı

İki tane dönüşüm yapan sınıf bulunmaktadır. Discrete fourier transform ve discrete cosine transform –II dönüşümlerini yapmak için dosyadan değerler okunacak ve bu değerlere bağlı olarak

dönüşümler yapılacak. Oluşan yeni değerler dosyaya yazdırılacaktır. Ek olarak bazen DFT’de dönüşümün çalışma süresi ekrana yazdırılacaktır.

4.2. Transform – Template Method Design UML Diyagramı



4.3. Transform – Template Method Design Sınıf Yapısı

Template method tasarım örünüüsüne uygun olarak ortak methodlar üst soyut sınıfta bulunmaktadır. Konsept gereği iki dönüşüm de dosyadan veri okuyacak ve onları dönüşümden sonra dosyaya yazacaktır. Ek olarak DFT’de kullanıcının isteğine bağlı olarak dönüşümün süresi de ekrana yazılacaktır. Bu sıra translate methodunda bulunmaktadır.

Exec ve hook methodları alt sınıflarda doldurulmaktadır. Exec methodunda dönüşüm yapılmaktadır. Hook ise yapılması istenen ek işi yapmaktadır. setOption ve getOption methodları ise kullanıcının DTF’de ekrana dönüşüm süresi ekrana basılması için yazıldı. Çalışma zamanında girmek istemediğim için böyle bir method yazdım.

4.4. Transform – Template Method Design Sonuçlar

```
1.1  
2.2  
3.3  
5.5  
4.4  
1.0
```

Dönüşüm yapılacak veri setidir.

```
1.0 0.0  
0.0 -0.8660254037844385  
0.0 0.49999999999999933  
0.0 3.061616997868383E-16  
0.0 -0.50000000000000013  
0.0 0.8660254037844392  
  
Process finished with exit code 0
```

DCT Sonucu çıktısı.

```
"C:\Program Files\Java\jdk1.8.0_191\bin\java.exe" ...  
17.5 0.0  
-6.6500000000000002 -0.0866025403784445  
1.1500000000000008 -1.9918584287042118  
0.10000000000000009 -1.7151244994428823E-15  
1.1499999999999977 1.9918584287042047  
-6.649999999999992 0.08660254037843229  
Would you like to see time of execution of Discrete Cosine Transform? [y/n]  
11
```

DFT Sonucu çıktısı ve kullanıcıya sorulacak dönüşüm süresi parametresi true verilmiştir. Milisaniye cinsinden sonuç ekrana yazıldı.