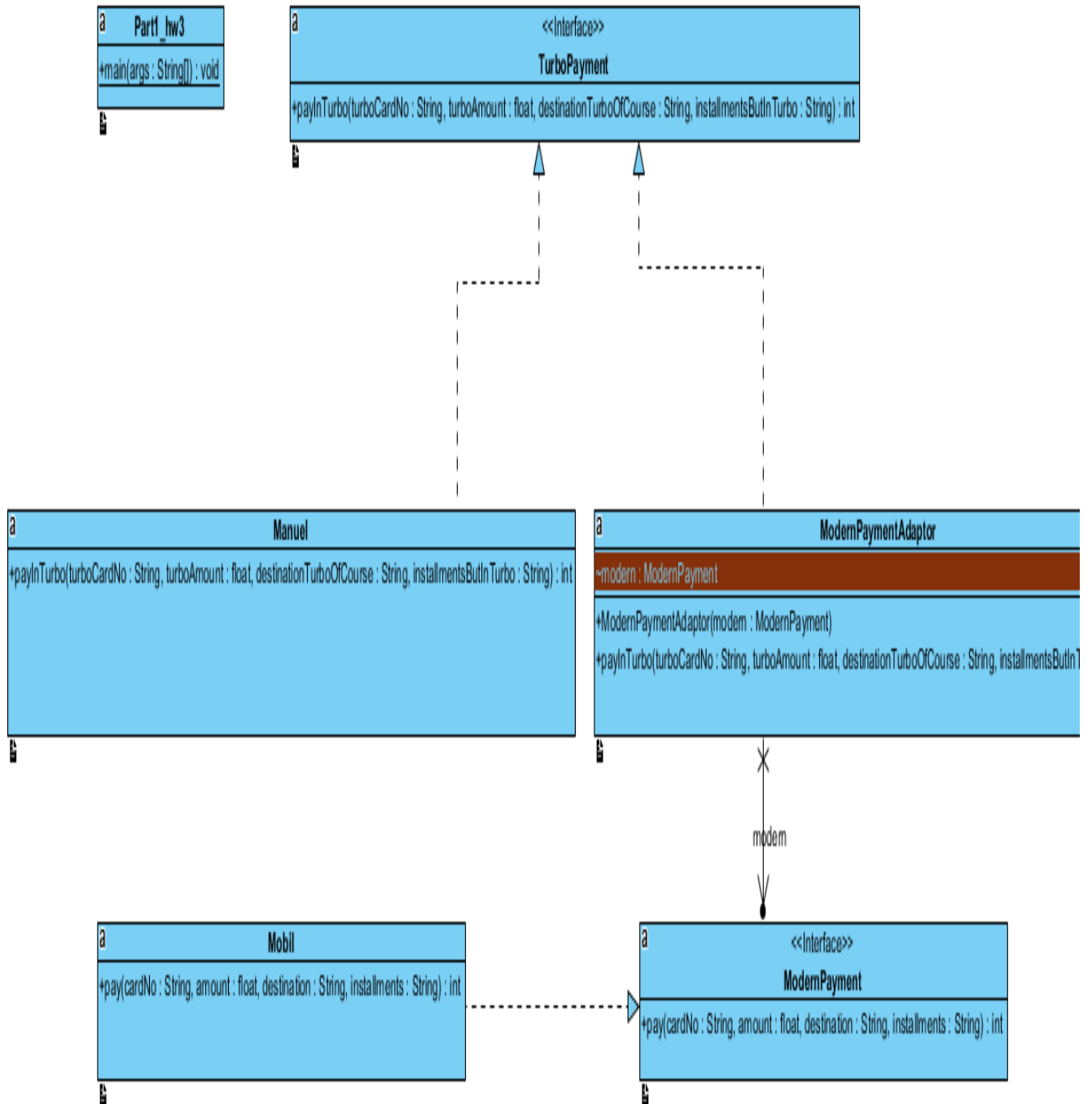


# CSE443 -Object Oriented Analysis and Design

## ~ RAPOR 3 ~

### PART1:

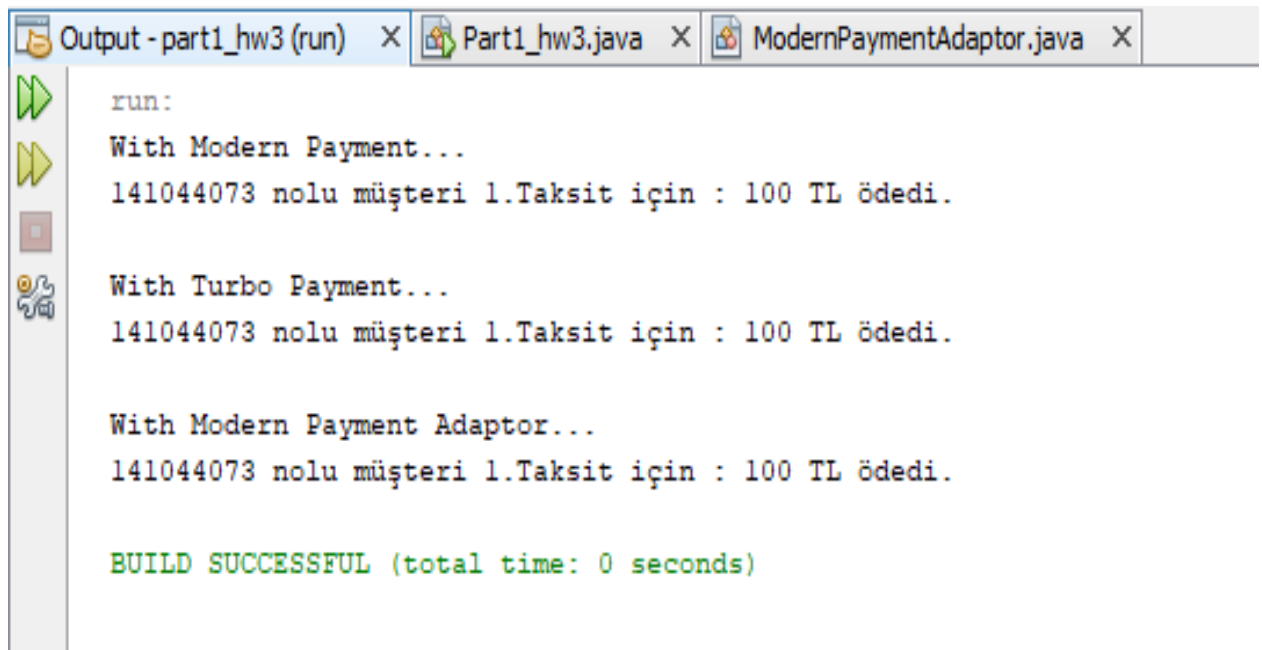
### Class Diagram



## Tasarım Açıklaması

- Bu bölümde problemi çözmek için **Adapter** Design Pattern kullanılarak,gerekli class ve classlar arası ilişki gerçekleştirildi.
- Tasarım olarak TurboPayment adında bir interface tanımlandı.
- Bu interfacede payInTurbo(--)) adlı methodun signature tanımlanmıştır.
- ModernPayment adlı interface ise pay(--)) adlı methoda sahiptir.
- ModernPaymentAdaptor adlı classımız TurboPayment implement ederek içeride ModernPayment objesi tutarak payInTurbo() içerisinde pay() çağrılır.

## Test Demo



```
run:
With Modern Payment...
141044073 nolu müşteri 1.Taksit için : 100 TL ödedi.

With Turbo Payment...
141044073 nolu müşteri 1.Taksit için : 100 TL ödedi.

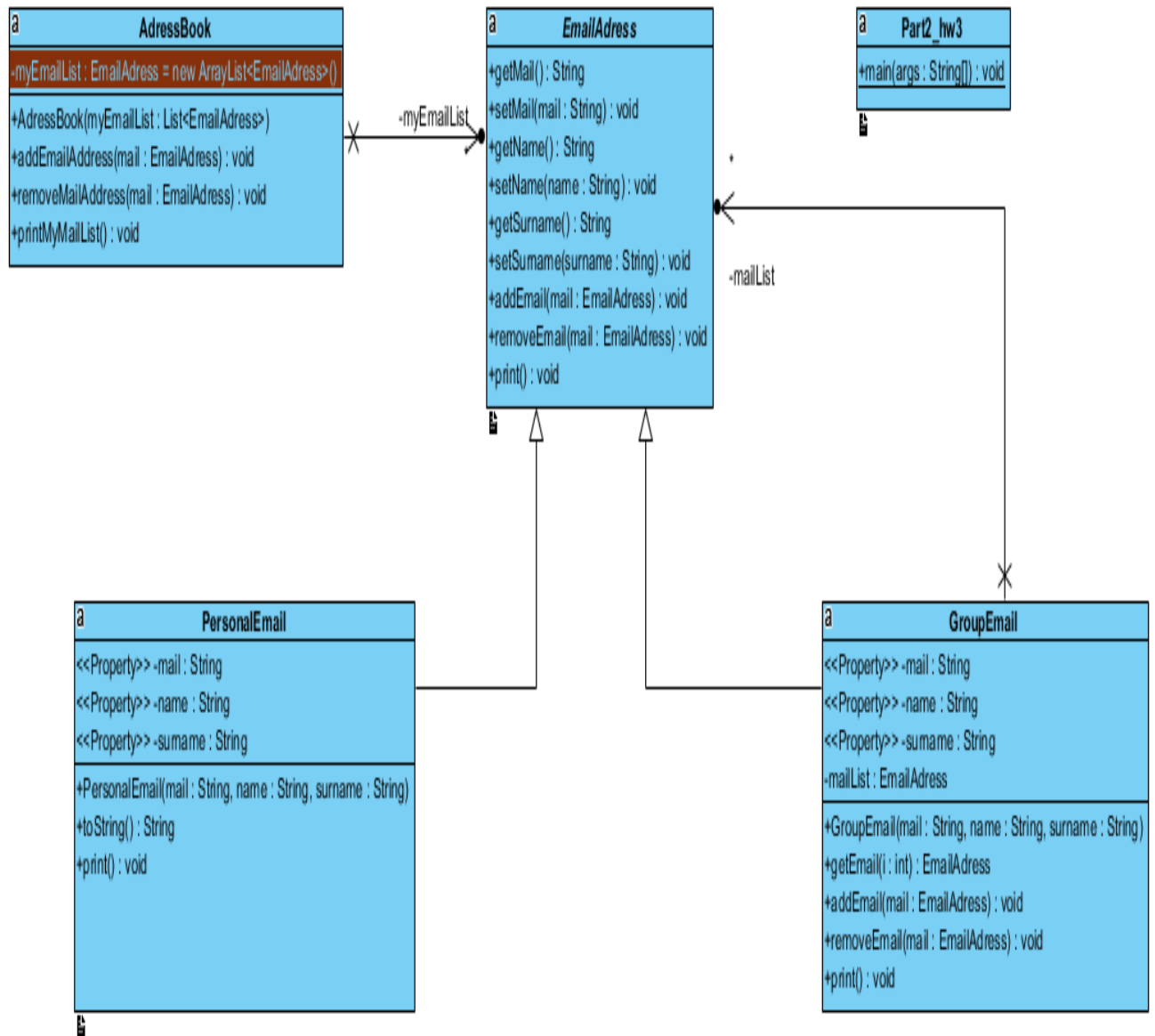
With Modern Payment Adaptor...
141044073 nolu müşteri 1.Taksit için : 100 TL ödedi.

BUILD SUCCESSFUL (total time: 0 seconds)
```

- ➔ ModernPayment mobil = new Mobil();
  - ➔ TurboPayment manuel = new Manuel();
  - ➔ TurboPayment modernAdapter = new ModernPaymentAdaptor(mobil);
- Şeklinde objeler kullanılarak test edilmiştir.

## PART2:

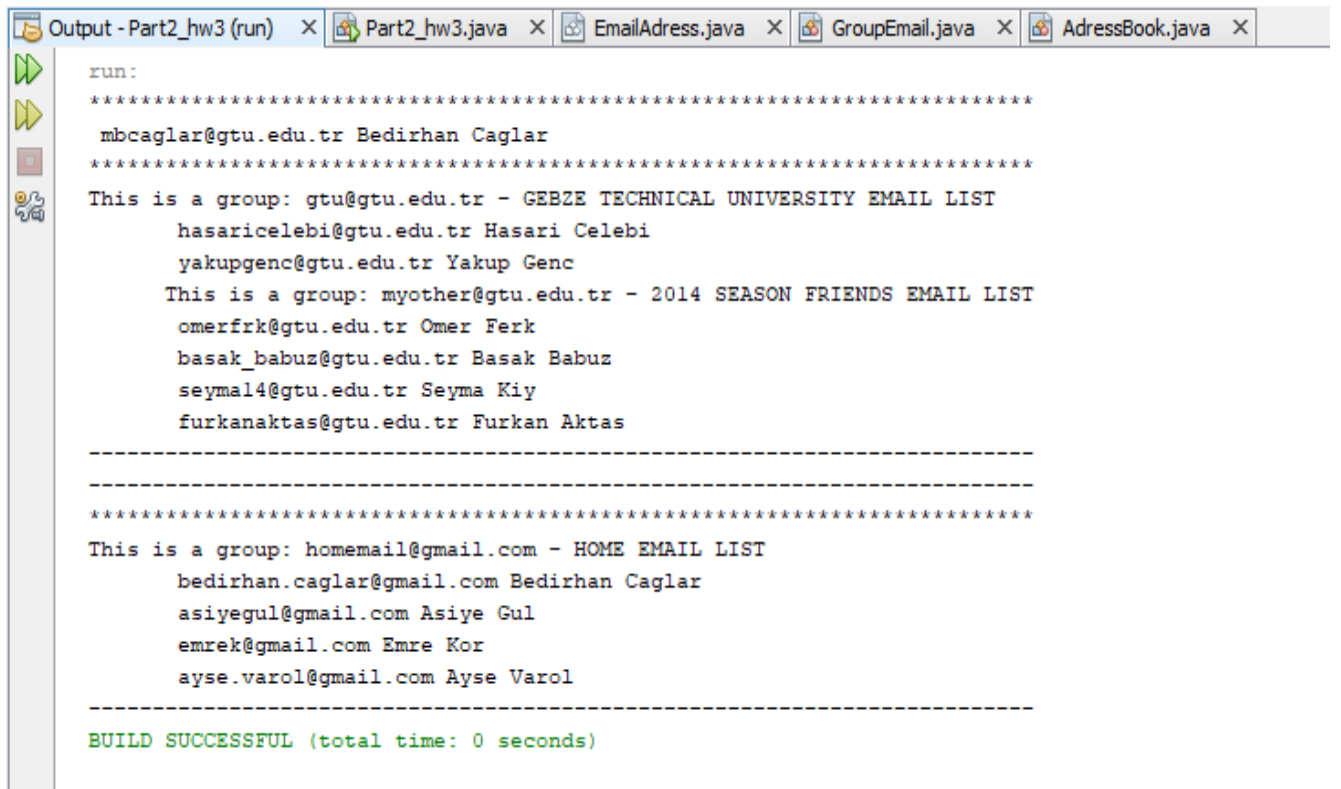
### Class Diagram



## Tasarım Açıklaması

- Bu bölümde problemi çözmek için **Composite** Design Pattern kullanılarak,gerekli class ve classlar arası ilişki gerçekleştirildi.
- Tasarım olarak EmailAdress abstract classı tanımlandı.
- Bu abstract classı ise PersonalEmail ve GroupEmail extends ederek gerekli methodları implement etti.
- GroupEmail classı içerisinde EmailAdress Listesi tutarak composite yapı oluşturuldu.
- AdressBook classı tanımlandı.

## Test Demo

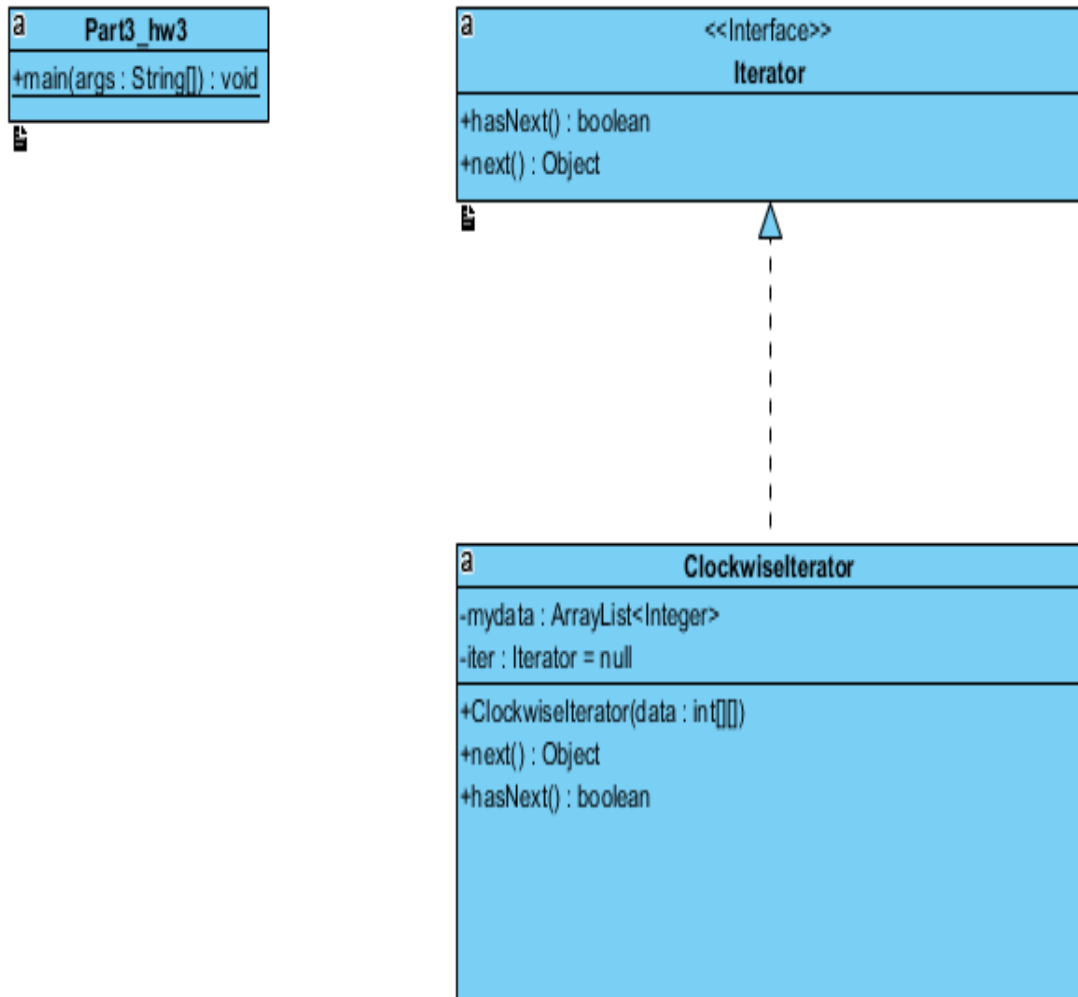


```
run:
*****
mbcaglar@gtu.edu.tr Bedirhan Caglar
*****
This is a group: gtu@gtu.edu.tr - GEBZE TECHNICAL UNIVERSITY EMAIL LIST
  hasaricelebi@gtu.edu.tr Hasari Celebi
  yakupgenc@gtu.edu.tr Yakup Genc
This is a group: myother@gtu.edu.tr - 2014 SEASON FRIENDS EMAIL LIST
  omerfrk@gtu.edu.tr Omer Ferk
  basak_babuz@gtu.edu.tr Basak Babuz
  seymal4@gtu.edu.tr Seyma Kiy
  furkanaktas@gtu.edu.tr Furkan Aktas
-----
*****
This is a group: homemail@gmail.com - HOME EMAIL LIST
  bedirhan.caglar@gmail.com Bedirhan Caglar
  asiye Gul
  emrek@gmail.com Emre Kor
  ayse.varol@gmail.com Ayse Varol
-----
BUILD SUCCESSFUL (total time: 0 seconds)
```

- ➔ Mainde personal ve group emaileri tanımlanarak adressbook üzerinden listeye eklenerek print edilmiştir.

## PART3:

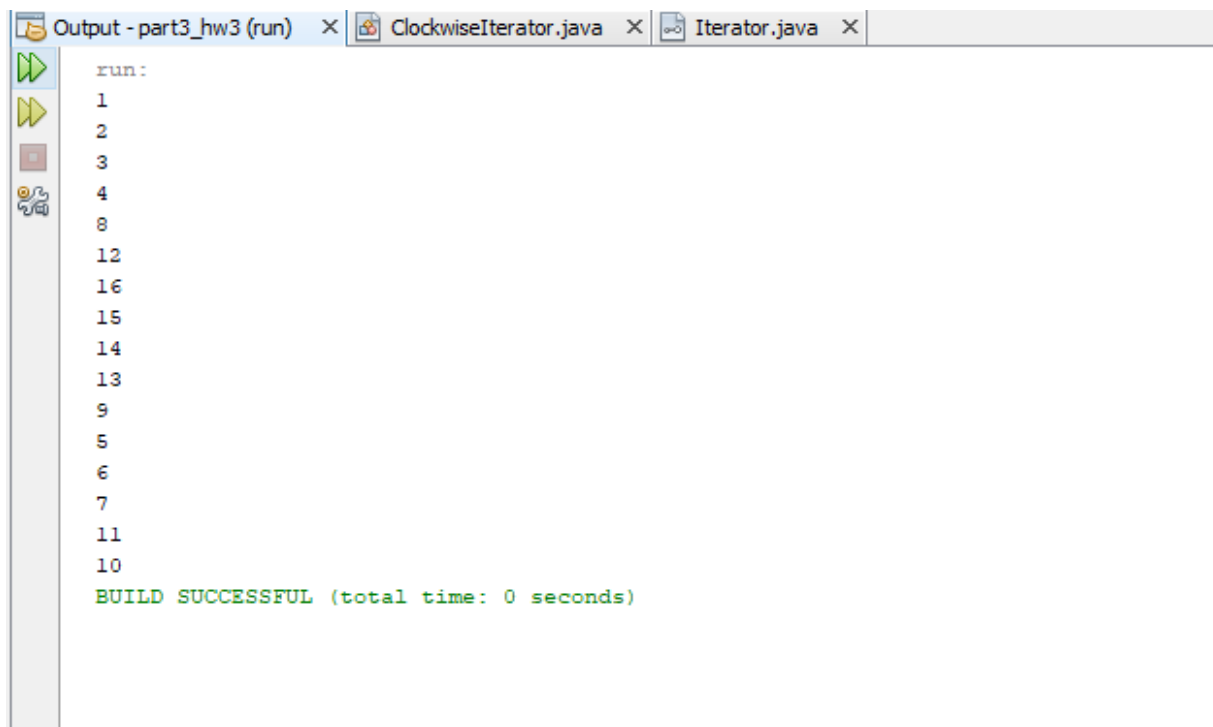
### Class Diagram



## Tasarım Açıklaması

- Bu bölümde problemi çözmek için **Iterator** Design Pattern kullanılarak,gerekli class ve classlar arası ilişki gerçekleştirildi.
- Tasarım olarak ClockwiseIterator classı ve Iterator interface tanımlandı.
- hasNext() ve next() methodları implement edilmek üzere ClockwiseIterator bırakıldı.
- ClockwiseIterator içerisinde ArrayList tutarak ,verilen 2D array spirally clockwise şeklinde assign edildi.
- Member olan ArrayListin iteratorü class içerisindeki itere assign edilerek hasNext ve next methodları gerçekleştirildi.

## Test Demo

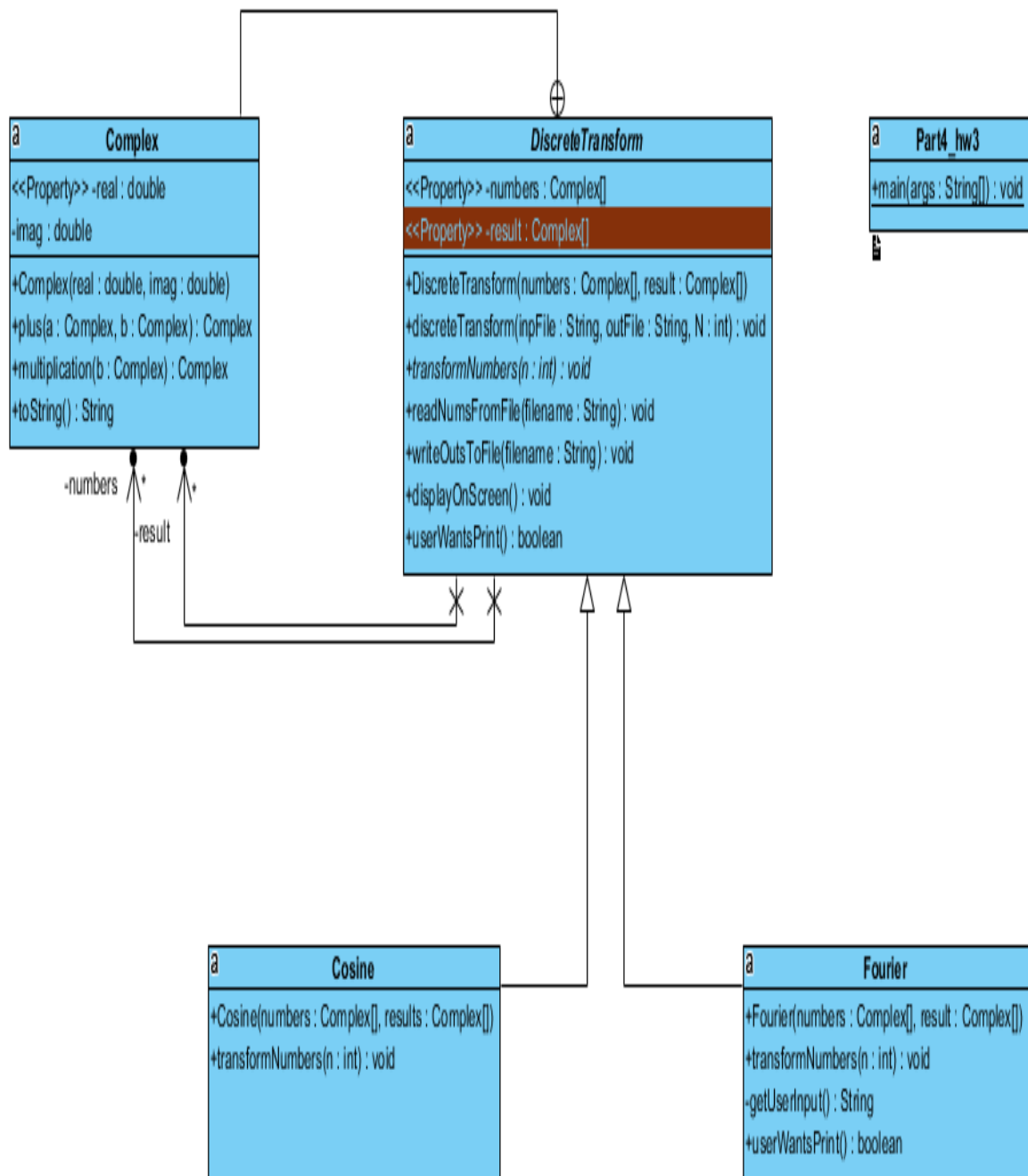


```
Output - part3_hw3 (run) x ClockwiseIterator.java x Iterator.java x
run:
1
2
3
4
8
12
16
15
14
13
9
5
6
7
11
10
BUILD SUCCESSFUL (total time: 0 seconds)
```

- ➔ Mainde int arr[][]={{1,2,3,4},{5,6,7,8},{9,10,11,12},{13,14,15,16}} tanımlanarak test edildi.

## PART4:

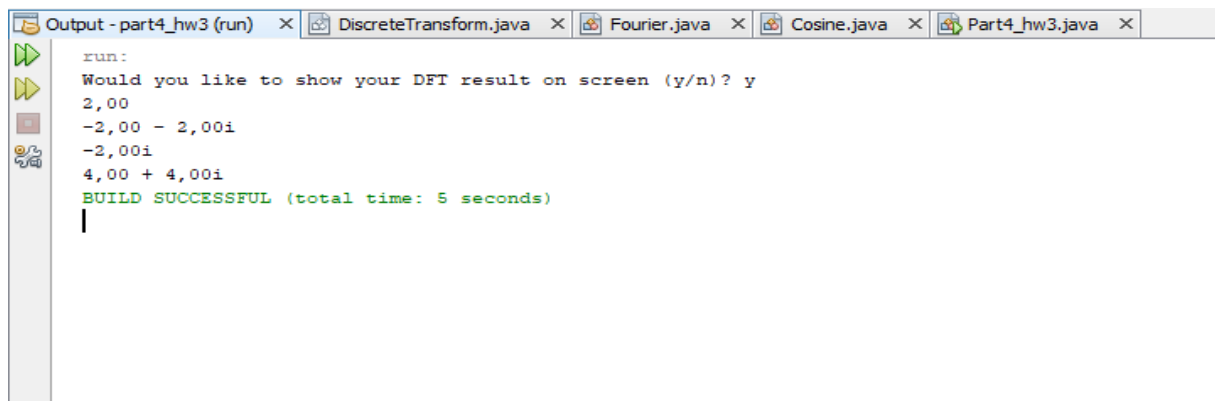
### Class Diagram



## Tasarım Açıklaması

- Bu bölümde problemi çözmek için **TemplateDesign** Pattern kullanılarak,gerekli class ve classlar arası ilişki gerçekleştirildi.
- Tasarım olarak DiscreteTransform abstract classı tanımlanarak içerisinde Complex static classının numbers ve result objeleri array yapısında tutuldu.
- Cousine ve Fourier classları bu abstract classı extend ederek implementaionları gerçekleştirdi.
- input adlı dosya okunarak işlemler gerçekleştirildi .
- Kullanıcıya sonuçları ekranda görmek isteyip istemediği sorularak outputF ve outputC dosyalarına yazıldı.

## Test Demo



```
run:
Would you like to show your DFT result on screen (y/n)? y
2,00
-2,00 - 2,00i
-2,00i
4,00 + 4,00i
BUILD SUCCESSFUL (total time: 5 seconds)
```

Main :

```
public static void main(String[] args) {
    // TODO code application logic here
    {
        DiscreteTransform.Complex[] cmplex = new DiscreteTransform.Complex[10];
        DiscreteTransform.Complex[] cmplex2 = new DiscreteTransform.Complex[10];

        DiscreteTransform dft = new Fourier(cmplex, cmplex2);

        dft.discreteTransform("input", "outputF", 4);
    }

    {
        DiscreteTransform.Complex[] cmplex = new DiscreteTransform.Complex[10];
        DiscreteTransform.Complex[] cmplex2 = new DiscreteTransform.Complex[10];

        DiscreteTransform dct = new Cosine(cmplex, cmplex2);

        dct.discreteTransform("input", "outputC", 4);
    }
}
```