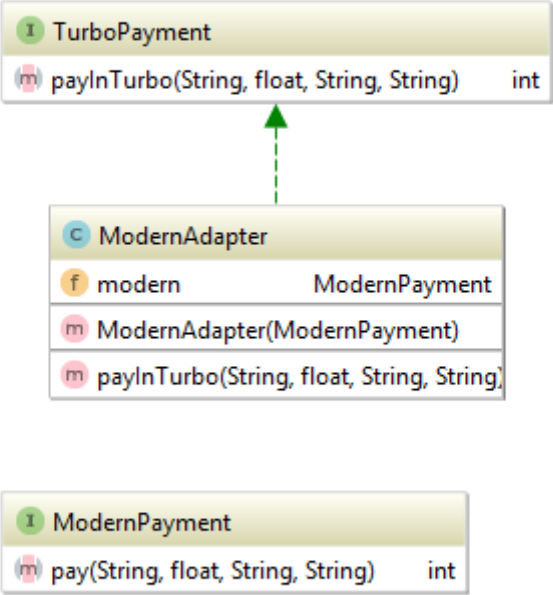


1-)

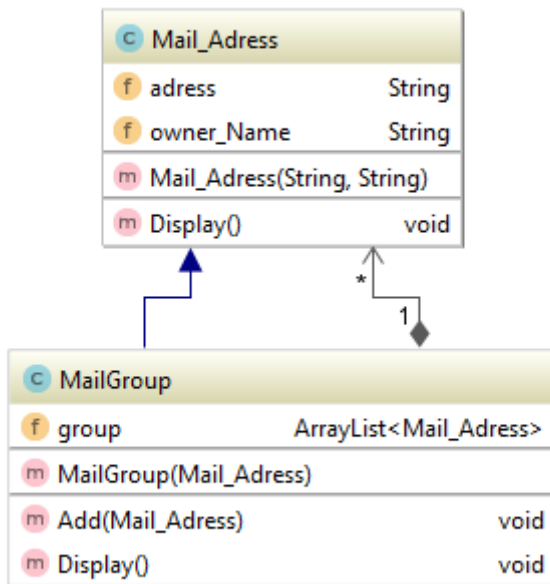


```
Main.java x TurboPayment.java x ModernAdapter.java x ModernPayment.java x Package null x

1  /**
2   * Created by ORHAN on 9.12.2018.
3   */
4  public class ModernAdapter implements TurboPayment {
5      ModernPayment modern;
6
7      ModernAdapter(ModernPayment m) { modern=m; }
8      @Override
9      public int payInTurbo(String turboCardNo, float turboAmount, String destinationTurboOfCourse, String installmentsButInTurbo) {
10         modern.pay(turboCardNo,turboAmount,destinationTurboOfCourse,installmentsButInTurbo);
11         return 0;
12     }
13 }
```

It is self explanatory.No nned to explain any further.

2-)



```
public class MailGroup extends Mail_Address {
    ArrayList<Mail_Address> group;

    MailGroup(Mail_Address ad){
        super(ad.address,ad.owner_Name);

        group=new ArrayList<Mail_Address>();
    }
    void Add(Mail_Address mail){
        group.add(mail);
    }
    public void Display(){
        super.Display();
        for (int i = 0; i <group.size() ; i++) {
            group.get(i).Display();
        }
    }
}

public class Mail_Address {
    private String address;
    private String owner_Name;

    public Mail_Address(String ad,String owner){
        address=ad;
        owner_Name=owner;
    }

    public void Display(){
        System.out.println("Mail address: "+address);
        System.out.println("Owner : "+owner_Name);
    }
}
```

Adress Group class.

Mail class.

It is self explanatory.No nned to explain any further.

```

public static void main(String[] args) {

    Mail_Address mail=new Mail_Address( ad: "orhancelik@gmail.com", owner: "orhan çelik");
    Mail_Address mail_2=new Mail_Address( ad: "ahmetcelik@gmail.com", owner: "ahmet çelik");
    MailGroup grup_A=new MailGroup(new Mail_Address( ad: "grup_A@gmail.com", owner: "Grup_A"));
    MailGroup grup_B=new MailGroup(new Mail_Address( ad: "grup_B@gmail.com", owner: "Grup_B"));
    grup_A.Add(mail);
    grup_A.Add(mail_2);
    grup_B.Add(grup_A);
    mail.Display();
    System.out.println("////////////////////////////////");
    mail_2.Display();
    System.out.println("////////////////////////////////");
    grup_A.Display();
    System.out.println("////////////////////////////////");
    grup_B.Display();
}

```

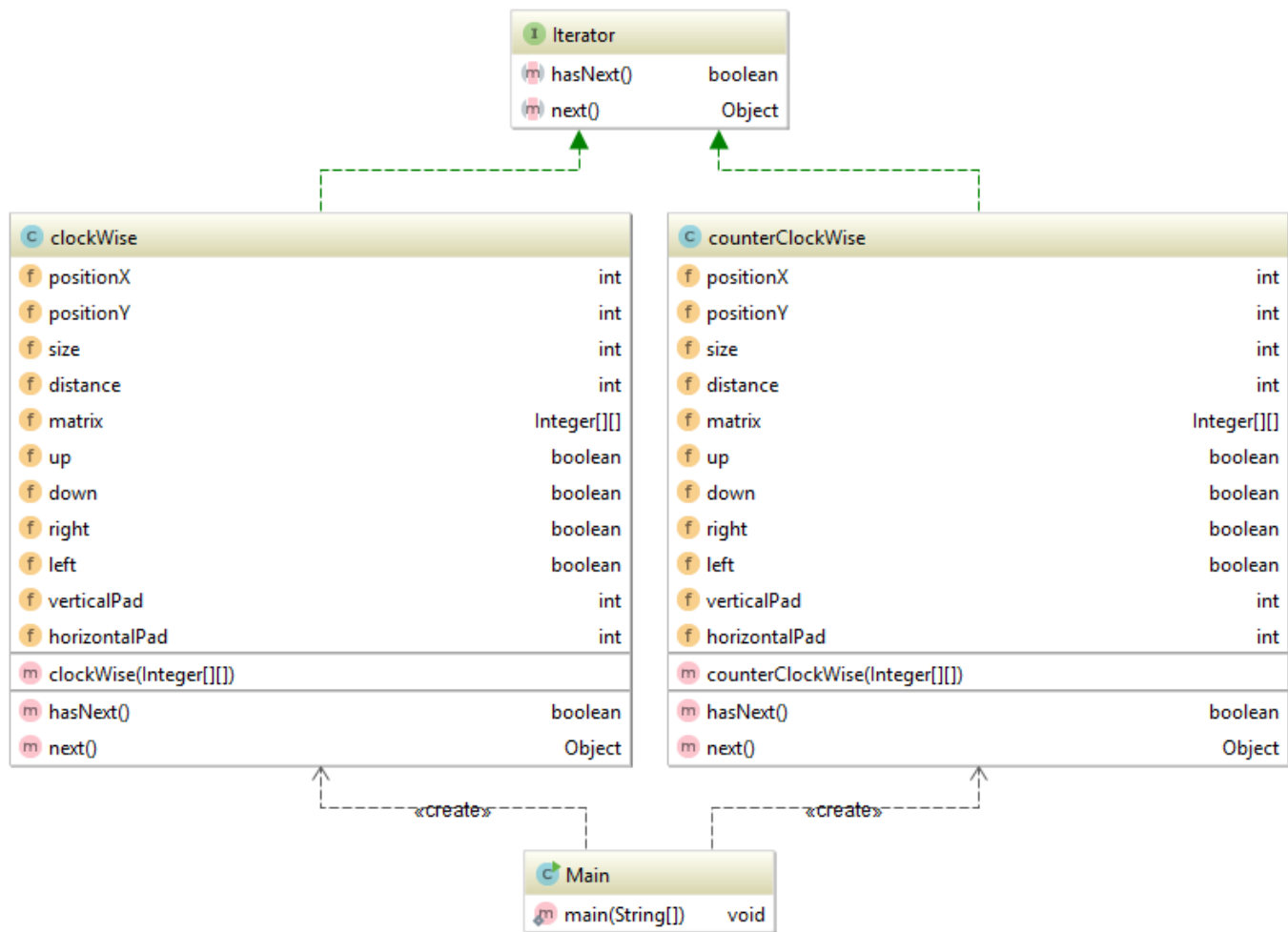
```

Mail adress: orhancelik@gmail.com
Owner : orhan çelik
////////////////////////////////
Mail adress: ahmetcelik@gmail.com
Owner : ahmet çelik
////////////////////////////////
Mail adress: grup_A@gmail.com
Owner : Grup_A
Mail adress: orhancelik@gmail.com
Owner : orhan çelik
Mail adress: ahmetcelik@gmail.com
Owner : ahmet çelik
////////////////////////////////
Mail adress: grup_B@gmail.com
Owner : Grup_B
Mail adress: grup_A@gmail.com
Owner : Grup_A
Mail adress: orhancelik@gmail.com
Owner : orhan çelik
Mail adress: ahmetcelik@gmail.com
Owner : ahmet çelik

```

Here is a working demo.

3-)



```

public static void main(String[] args) {
    System.out.println("Hello World!");
    Integer [][]matrix={{1,2,3,4},{5,6,7,8},{9,10,11,12},{13,14,15,16}};

    clockWise clock=new clockWise(matrix);

    while (clock.hasNext())
        System.out.print(clock.next()+" ");
    System.out.println();
    System.out.println("////////////////////////////////");
    counterClockWise counter=new counterClockWise(matrix);
    while (counter.hasNext())
        System.out.print(counter.next()+" ");
}
  
```

```

1 2 3 4 8 12 16 15 14 13 9 5 6 7 11 10
////////////////////////////////
1 5 9 13 14 15 16 12 8 4 3 2 6 10 11 7
Process finished with exit code 0
  
```

Working demo.

```

public boolean hasNext() {
    if(distance<size)
        return true;
    return false;
}

@Override
public Object next() {
    distance++;

    if(down && matrix.length>positionX) {
        positionX++;
        if(matrix.length-horizontalPad/2-1 <=positionX) {
            down=false;
            right=true;
            verticalPad++;
        }
        return matrix[positionX][positionY];
    }
    else if(right && positionY<matrix[0].length) {
        positionY++;
        if(matrix[0].length-verticalPad<=positionY) {
            right=false;
            up=true;
            horizontalPad++;
        }
        return matrix[positionX][positionY];
    }

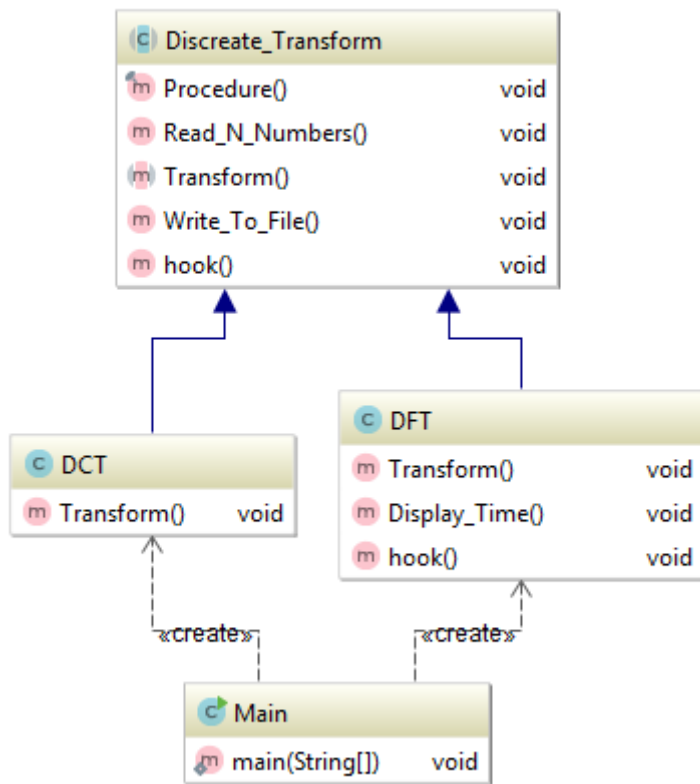
    else if(up && positionX>=0) {
        positionX--;
        if(positionX<=horizontalPad-1) {
            up=false;
            left=true;
        }
        return matrix[positionX][positionY];
    }
    if(left && 0<=positionY) {
        positionY--;
        if(positionY<=verticalPad) {
            down=true;
            left=false;
            horizontalPad++;
        }
        return matrix[positionX][positionY];
    }
    return null;
}

```

Code work with booleans to show the direction and Horizontal and Vertical paddings to show where to stop.

Only if else blocks changes places between them.

4-)



```

public abstract class Discreate_Transform {

    final public void Procedure() {
        Read_N_Numbers();
        Transform();
        Write_To_File();
        hook();
    }

    public void Read_N_Numbers() {
        System.out.println("Reading File");
    }

    abstract void Transform();

    public void Write_To_File() {
        System.out.println("Writing to File");
    }

    void hook() {
    }
}
  
```

Hook is for methods like displayin execution time.Things that user might want to add later by overriting this method.

Procedure method is Final.

```

public class DFT extends Discreate_Transform{

    @Override
    void Transform() {
        System.out.println("DFT transform");
    }

    void Display_Time(){
        System.out.println("Execution time");
    }

    @Override
    void hook() {
        Display_Time();
    }
}

```

Display_Time attached to hook.

```

public static void main(String[] args) {
    //System.out.println("Hello World!");

    Discreate_Transform dft=new DFT();
    Discreate_Transform dct=new DCT();

    dft.Procedure();
    System.out.println("////////////////////");
    dct.Procedure();
}

```

```

main
"C:\Program Files\Java\jdk1.8.0_151\bin\j
Reading File
DFT transform
Writing to File
Execution time
////////////////////
Reading File
DCT transform
Writing to File

Process finished with exit code 0

```

Working demo for design representation. Not fully implemented yet.