

```

public class Vol {
private Date dateDepart,dateArrivee;
private Time heureDepart,heureArrivee;
private Aeroport aeroportDepart,AeroportArrivee;
private EtatVol etatVol;
private Vector<Escale>listeEcales;
private Vector<Reservation>listeReservations;
private Compagnie compagnie;
public Vol(Compagnie c,Date dateDepart,Date dateArrivee,Time heureDepart,Time
heureArrivee,Aeroport aeroportDepart,Aeroport AeroportArrivee)
{this.compagnie=c; // this est facultative ici
this.dateArrivee=dateArrivee;
this.dateDepart=dateDepart;
this.heureArrivee=heureArrivee;
this.heureDepart=heureDepart;
this.aeroportDepart=aeroportDepart;
this.AeroportArrivee=AeroportArrivee;
listeEcales =new Vector<Escale>();
listeReservations=new Vector<Reservation>();
etatVol=EtatVol.AHEURE;//constante AHEURE de l'ensemble <enum>

```

```

}
public void ajouterEscale(Aeroport aeroport,int duree ,Time heureDepart,Time
heureArrivee)
{ Escale e=new Escale(aeroport,duree,heureDepart,heureArrivee);
listeEcales.add(e);
}

//getter et Setter
public void afficher()
{ System.out.println(" "+this.aeroportDepart.getNom()+
" "+this.dateDepart+
" "+this.heureDepart+
" "+this.AeroportArrivee.getNom()+
// ajouter les autres champs
" "+this.etatVol.toString();//print peut imprimer l'objet
sans toString
);
for (Escale e:this.listeEcales)
{ e.afficher();
}
for (Reservation r:this.listeReservations)
{ r.afficher();
}
}
void reserver(int i) {
{Reservation r=new Reservation(i,this);// son vol est this
this.listeReservations.add(r);
} }}

```

```

public class Reservation {
private Date dateReservation;
    private int num;
    private Vol vol;
    private boolean etatRes;//ou avec enum
    public Reservation(int n,Vol v)
    {
        dateReservation=Calendar.getInstance().getTime();
        num=n;
        vol=v;
        etatRes=false;
    }
    public void confirmer()
    {etatRes=true;}

    //getter et setter

    void afficher() {
// à remplir
    }
}

```

```

public class Aeroport {
private String nom;
    Ville ville;
    public Aeroport(Ville ville,String nom)
    {this.nom=nom;
    this.ville=ville;}
public String getNom() {return nom;}
public void setNom(String nouveauNom){this.nom=nouveauNom;}

    public Ville getVille() {return ville;}
    public void setVille(Ville nouvelleVille){this.ville=ville;}

void afficher() {
    // à remplir
}

```

```
public class Compagnie {  
private String nom;  
private int num;  
public Compagnie (String nom,int num)  
    {this.nom=nom;  
    this.num=num;  
    }  
    //getter et setter  
    //afficher  
}
```

```
public class Escale {  
private Aeroport aeroport;  
private int duree ;  
private Time heureDepart,heureArrivee;  
public Escale(Aeroport aeroport,int duree ,Time heureDepart,Time heureArrivee)  
{ this.aeroport=aeroport;  
  this.duree=duree;  
  this.heureArrivee=heureArrivee;  
  this.heureDepart=heureDepart;  
}  
  
void afficher() {  
    // à remplir  
}  
}
```

```
public enum EtatVol {
    ANNULE,
    RETARDE,
    ARCHIVE,
    AHEURE;
}
```

```
import java.sql.Date;
import java.sql.Time;
public class Gestion {
public static void main(String[] args) {
    Compagnie c1=new Compagnie("airAlgérie",1);
    Ville v1,v2,v3;
    v1=new Ville("Oran");
    v2=new Ville("Constantine");
    v3=new Ville("Alger");
v1.ajouterAeroport("ahmed BenBella");
v2.ajouterAeroport("Mohamed Boudiaf");
v3.ajouterAeroport("Boulmediane");
    Aeroport a1,a2;
    a1 =v1.getListAeroport().get(0);// le premier aeroport de la ville Oran
    a2=v2.getListAeroport().get(0);// le premier aeroport de constantine
    Date d1;
    d1=Date.valueOf("2016-02-02");
    Time t1,t2,t3,t4;
    t1=Time.valueOf("08:00:00");t2=Time.valueOf("11:00:00");
    Vol voll = new Vol(c1,d1,d1,t1,t2,a1,a2);
```

```
voll.ajouterEscale(v3.getListAeroport().get(0)
,1,Time.valueOf("09:00:00"),Time.valueOf("10:00:00")); //les declarer avant,
possible
voll.afficher();
    voll.reserver(12);
    voll.reserver(13);
voll.afficher();
    //voll.getListReservation().get(1).confirmer();

    } }
```

Ex1

