

```

public class CmpEco extends ComportementConcret {

    @Override
    public double getPrix(int km) {
        return km*0.1;
    }

    @Override
    public int getDelai() {
        return 3;
    }
}

public class CmpFroid extends ComportementConcret {

    @Override
    public double getPrix(int km) {
        int majoration = km/100;
        return km*0.25+majoration;
    }

    @Override
    public int getDelai() {
        return 2;
    }
}

public class CmpUrgent extends ComportementConcret {

    private static final double prixZ1 = 0.5;
    private static final double prixZ2 = 1;
    private static final int zone1 = 30;

    @Override
    public double getPrix(int km) {
        double prix = 0;
        if(km<zone1){
            prix=km*prixZ1;
        }
        else{
            prix=km*prixZ2;
        }
        return prix;
    }

    @Override
    public int getDelai() {
        return 1;
    }
}

public interface ComportementColis {
    public double getPrix(int km);
    public int getDelai();
}

```

```

public abstract class ComportementConcret implements
ComportementColis, Comparable<ComportementColis> {

    public boolean equals(Object o){
        return this.getClass().equals(o.getClass());
    }
    public int hashCode(){
        return this.getClass().getSimpleName().hashCode();
    }

    public int compareTo(ComportementColis c){
        return
this.getClass().getSimpleName().compareTo(c.getClass().getSimpleName());
    }

    public String toString(){return this.getClass().getSimpleName();}
}

public class MoyenTransport {
    private List<ComportementColis> comportementsGeres = new
ArrayList<ComportementColis>();
    private String immatriculation;
    private List<Colis> chargement = new ArrayList<Colis>();

    public MoyenTransport(String immatriculation, List<ComportementColis> cmp){
        this.immatriculation = immatriculation;
        comportementsGeres = cmp;
    }

    public boolean addColis(Colis c){
        boolean ret = false;
        if(comportementsGeres.containsAll(c.getCaracteristiques())){ret =
chargement.add(c);}
        return ret;
    }

    public List<ComportementColis> getComportementsGeres(){return
comportementsGeres;}

    public String toString(){
        String ret = "Chargement du véhicule "+immatriculation;
        for(Colis c:chargement){
            ret+="\n\t"+c;
        }
        return ret;
    }
}

```

```

public class Agence {
    private String nom;
    private String ville;
    private Set<ComportementColis> comportementsPossibles = new
TreeSet<ComportementColis>();
    private List<MoyenTransport> transports;

    private List<Colis> lesColis = new ArrayList<Colis>();

    public Agence(String nom, String adresse, List<MoyenTransport> transports){
        this.nom=nom;
        this.ville=adresse;
        this.transports=transports;
        if(transports!=null){majComportements();}
    }

    public Agence(String nom, String adresse){
        this(nom, adresse, null);
        transports = new ArrayList<MoyenTransport>();
    }

    private void majComportements() {
        for(MoyenTransport m:transports){
            comportementsPossibles.addAll(m.getComportementsGeres());
        }
    }

    public boolean addColis(Colis c){
        boolean ok=true;
        Iterator<ComportementColis> it = c.getCaracteristiques().iterator();
        while(ok && it.hasNext()){
            if(!comportementsPossibles.contains(it.next())){
                ok=false;
            }
        }
        if(ok){lesColis.add(c);}
        return ok;
    }

    public void affecteChargement(){
        for(Colis c:lesColis){
            boolean trouve=false;
            Iterator<MoyenTransport> it = transports.iterator();
            while(!trouve && it.hasNext()){
                if(it.next().addColis(c)){trouve=true;}
            }
        }
    }

    public double calculPrix(int km){
        double prix=0;
        for(Colis c:lesColis){prix+=c.getPrix(km);}
        return prix;
    }

    public String toString(){
        String cmp="";
        for(ComportementColis cc:comportementsPossibles){
            cmp+="\t"+cc+"\n";
        }
        return "Agence "+nom+ " située à "+ville+"\n"+cmp;
    }
}

```

```

public class Colis implements ComportementColis{
    private static int numeroCourant = 1;

    private List<ComportementColis> caracteristiques = new
ArrayList<ComportementColis>();
    private int numero;

    public Colis(List<ComportementColis> caracteristiques){
        numero = Colis.numeroCourant;
        Colis.numeroCourant++;
        this.caracteristiques = caracteristiques;
    }

    public boolean hasComportement(ComportementColis cc) {
        return caracteristiques.contains(cc);
    }

    public List<ComportementColis> getCaracteristiques() {
        return caracteristiques;
    }

    public String toString(){
        String info="";
        for(ComportementColis cc:caracteristiques){info+=cc+",";}
        return "Colis n°"+numero+" (" +info+")";
    }

    @Override
    public double getPrix(int km) {
        double ret=0;
        for(ComportementColis c:caracteristiques)
        {
            ret+=c.getPrix(km);
        }
        return ret*km;
    }

    @Override
    public int getDelai() {
        int ret=0;
        for(ComportementColis c:caracteristiques)
        {
            if(ret>0 && c.getDelai()<ret){ret = c.getDelai();}
            else{ret = c.getDelai();}
        }
        return ret;
    }
}

```

```

public class Test {

    public static List<Colis> fabriqueColis(int nb){
        List<Colis> liste = new ArrayList<Colis>();
        for (int i=0;i<nb;i++){
            List<ComportementColis> cc = new ArrayList<ComportementColis>();
            if(Math.random()<0.5){cc.add(new CmpUrgent());}
            if(Math.random()<0.5){cc.add(new CmpFroid());}
            cc.add(new CmpEco());
            liste.add(new Colis(cc));
        }
        return liste;
    }

    /**
     * @param args
     */
    public static void main(String[] args) {
        ComportementColis c1 = new CmpUrgent();
        ComportementColis c2 = new CmpFroid();
        ComportementColis c3 = new CmpEco();

        List<ComportementColis> lT1=new ArrayList<ComportementColis>();
        List<ComportementColis> lT2=new ArrayList<ComportementColis>();
        List<ComportementColis> lT3=new ArrayList<ComportementColis>();

        lT1.add(c1);
        lT1.add(c3);
        lT2.add(c3);
        lT3.add(c1);
        lT3.add(c3);
        List<MoyenTransport> transports = new ArrayList<MoyenTransport>();
        transports.add(new MoyenTransport("1234FZ21",lT1));
        transports.add(new MoyenTransport("4578AZ12",lT2));
        transports.add(new MoyenTransport("BF214",lT3));

        Agence a = new Agence("Mon Agence","Lyon", transports);
        System.out.println(a);

        List<Colis> l = fabriqueColis(20);
        for(Colis c:l){
            if(a.addColis(c)){System.out.println("Colis "+c +" ajouté");}
            else{System.out.println("Colis "+ c + "non pris en charge");}
        }

        a.affecteChargement();
        System.out.println("Prix du transport pour 25km: "+a.calculPrix(25));
        System.out.println("Prix du transport pour 185km: "+a.calculPrix(185));

    }

}

```