

# RAPPORT PROJET POO JAVA



Préparé par : KHOUMERI Lyes Mehdi | 191931065210  
NASS Manel | 191931079967

Date : 24 juin 2021

Dans ce projet, on s'intéresse à la gestion du réseau des différents moyens de transport urbain (métro, tramway, bus, train) sur London .

Le travail a été fait à travers l'implémentation de plusieurs méthodes sur plusieurs classes

## 1) Signature de méthodes :

### Partie I : Requêtes de mise à jours

#### Classe cord :

##### 1. Constructeur :

```
public Cord(double longitude, double latitude)
```

##### Explication :

Une méthode qui construit un objet de la classe .( **Explication s'applique sur toutes les classes** ).

##### 2. Getters :

```
public double getLongitude()  
. public double getLatitude()
```

##### Explication:

Les méthodes qui récupèrent les attributs en dehors de la classe car c'est des attributs privés. ( **Explication s'applique sur toutes les classes** ).

### 3. Setters :

```
public void setLongitude(double longitude)
public void setLatitude(double latitude)
```

#### **Explication:**

Les méthodes qui modifient les attributs en dehors de la classes car c'est des attributs privée.( **Explication s'appliquent sur toutes les classes** ).

### 4. ToString :

```
public String toString()
```

#### **Explication :**

Une méthode qui ( **Explication s'applique sur toutes les classes** ).

### 5. affichage :

```
public void affiche()
```

#### **Explication:**

Affiche les attributs de la classe.( **Explication s'appliquent sur toutes les classes** ).

### 5. Deg2rad :

```
static double deg2rad(double deg)
```

### Explication:

Méthode qui convertie du degré au radian

#### 6. Dist :

```
static double dist(Cord S1,Cord S2)
```

### Explication :

Une méthode qui calcule la distance entre deux points GPS.

## Classe MoyTransport :

#### 1. Getters :

```
public MoyType getType()  
. public void setType(MoyType type)
```

#### 2. Contructeur :

```
public MoyTransport(MoyType type)
```

#### 3. ToString :

```
public String toString()
```

## Declaration de type Enum :

```
public enum  
MoyType{ BUS(LocalTime.of(6,0),LocalTime.of(20,30),60,"\uD83D\uDE8C",30)  
,  
METRO(LocalTime.of(5,0),LocalTime.of(23, 0),200," ",50) ,  
TRAMWAY(LocalTime.of(5,30),LocalTime.of(22,0),150,"\uD83D\uDE8A",15)  
. TRAIN(LocalTime.of(6,0),LocalTime.of(21,0),300,"\uD83D\uDE82",90) ;
```

## Explication :

### 2. constructeur :

```
MoyType(LocalTime heureDebut, LocalTime heureFin, int nbrVoy, String  
emoji, int v)
```

### 3.Getters :

```
public int getNbrVoy()  
public LocalTime getHeureDebut()  
public LocalTime getHeureFin()  
public String getEmoji()  
public int getV()
```

### 4.ToString :

```
public String toString()
```

#### 4. Affiche :

```
public void affiche()
```

## Classe ligne :

#### 1. Constructeur :

```
public Ligne(String numligne, MoyTransport.MoyType TRANS, Station  
station_depart, Station station_arrive, ArrayList<Station> station_inter,  
ArrayList<Troncon> troncon)
```

#### 2. Getters et Setters :

```
public String getNumligne()  
public void setNumligne(String numligne)  
public Station getStation_depart()  
public void setStation_depart(Station station_depart)  
public void setStationDepartNOM(String nom)  
public void setStation_departttt(Station station_depart)  
public Station getStation_arrive()  
public void setStation_arrive(Station station_arrive)  
public void setStationArriveNOM(String  
nom){this.station_arrive.setNom(nom);}  
public void setStation_arrivettt(Station station_arrive)
```

```
public void setStation_inter(ArrayList station_inter)
public void setStation_interfff(Station s,int p)
public void setTroncon(ArrayList troncon)
public MoyTransport.MoyType getTRANS()
public ArrayList<Navette> getNavette()
public void setNavette(ArrayList<Navette> navette)
public void setNavette(Navette navette)
Static void setCoorespondance(Station s)
```

### 3.Delnavette :

```
public void DelNavette(Navette navette)
```

### **Explication :**

Methode qui supprime une navette de la ligne

### 5. ToString :

```
public String toString()
```

### 6. Affichlist :

```
public void affichelist()
```

### **Explication :**

### 7. affichelist2 :

```
public void affichelist2()
```

8. affiche :

```
public void affiche()
```

9. ajoutS :

```
public void ajoutS(Station S1)
```

**Explication :**

Methode pour ajouter une station dans une ligne .

9. CreatNav:

```
public void CreatNav()
```

10. LignePosition :

```
static int LignePosition (ArrayList<ArrayList<Ligne>> Lignes )
```

**Explication :**

Methode qui retourne la position d'une ligne .

11. VerifID :

```
static Boolean VerifID(String id ,ArrayList<ArrayList<Ligne>> Lignes,int ar)
```

**Explication :**

Une methode qui verifie si une ligne existe

12. indxNav :



```
public int indxNav(int num)
```

### **Explication :**

Une methode qui retourne l'indexe d'une navette depuis son numero

### 13. ReenumStat:

```
public void ReenumStat ()
```

### **Explication :**

Reenumerer toute les stations par ordre

## **Classe Navette (Fille de MoyTransport) :**

### 1 . Constructeur :

```
public Navette(MoyType type, int numero, String marque, int annee, Ligne  
ligne)
```

### 2 .Getters & Setters :

```
Public int getNumero()  
public void setNumero(int numero)  
public String getMarque()  
public void setMarque(String marque)
```

```
public int getAnnee()  
public void setAnnee(int annee)  
public Ligne getLigne()  
public void setLigne(Ligne ligne)  
public static ArrayList<Integer> getNavNumARRAY()  
public static void setNavNumARRAY(Integer n )
```

### 3. ToString:

```
public String toString()
```

### 4. Affichenav:

```
static void affichenav(ArrayList<ArrayList<Ligne>> Lignes)
```

## Classe Station :

### 1. Enum type:

```
public enum type{principale, secondaire};
```

### 2. constructeur:

```
public Station(int numero, String nom, Cord cordonee, type typestat,String s)  
public Station ()
```

### 3. Setters & Getters :

```
// Numero Getter & setters
```

```
public int getNumero()  
public void setNumero(int numero)
```

```
//+ nom Getter & setters
```

```
public String getNom()  
public void setNom(String nom)
```

```
//+ code unique
```

```
public int getCode()  
public void setCode(int code)
```

```
//+ cord Getter & setters
```

```
public Cord getCordonee()  
public void setCordonee(Cord cordonee)
```

```
//+ station Getter & setters
```

```

public type getStation()

public void setStation(type typestat)

// Type Getter & setters

public type getTypestat()
public void setTypestat(type typestat)

// array station principale

static ArrayList<Station> getArrayStatPrinc()

static void setArrayStatPrinc(Station s)

public String getNline()

. public void setNline(String nline)

public Ligne getLine()

public void setLine(Ligne line)

```

#### 4. Equals :

```

public boolean equals(Object o)

```

#### **Explication :**

Methode qui retourne si deux stations sont egales ( meme coordonnees spatiales )

#### 5. HashCode :

```

public int hashCode()

```

#### **Explication :**

6. Affiche :

```
public void affiche()
```

7. DupSt:

```
public boolean DupSt(ArrayList<Station> Ars)
```

**Explication :**

Methode qui verifie si une station est dupliqué dans une ligne

8. CreatStat :

```
static Station CreatStat(int num,String id)
```

**Explication :**

Une methode qui crée une station

9. Affihetoutestation :

```
static void afficheTouteStation(ArrayList<ArrayList<Ligne>> Lignes)
```

**Explication :**

une methode qui affiche toute les station d'une ligne

## 1. constructeur :

```
public Troncon(int numero, Station station_debut, Station station_fin, double kilometrage, etat state)
```

## 2. Getters and Setters :

```
//+ Num Getter & setters
```

```
public int getNumero()
```

```
public void setNumero(int numero)
```

```
//+ stationDebut Getter & setter
```

```
public Station getStation_debut()
```

```
public void setStation_debut(Station station_debut)
```

```
//+ stationFin Getter & setters
```

```
public Station getStation_fin()
```

```
public void setStation_fin(Station station_fin)
```

```
//+ kilometrage Getter & setters
```

```
public double getLongueur()
```

```
public void setLongueur(double kilometrage)
```

```
// station getter & setter
```

```
public etat getState()  
. public void setState(etat state)
```

3. affiche :

```
public void affiche()
```

4. Fusion :

```
public Troncon fusion(Troncon troncon2)
```

**Explication :**

Methode qui fait la fusion de deux tronçons

5. etatRand :

```
static Troncon.etat etatRand ()
```

**Explication :**

Une methode qui génère aleatoirement l'etat du tançons

6. StatToTronc :

```
static void StatToTronc(ArrayList<Station> ARstat, ArrayList<Troncon> ARtronc)
```

### **Explication :**

Methode qui transforme un array de stations a un array de tronçon

#### 7. dureeTronc :

```
public double dureeTronc (MoyTransport.MoyType M)
```

### **Explication :**

Methode qui calcule la durée estimée dans un trançon specifique

## Classe prog :

Classe contenant les manipulation de toutes les requêtes mise a jour

#### 1. AffichLignes :

```
static void affichLignes(ArrayList<ArrayList<Ligne>> Lignes)
```

### **Explication :**

Une méthode qui affiche toute les lignes

#### 2. CreatLigne :

```
static void CreatLigne(ArrayList<ArrayList<Ligne>> Lignes)
```

### **Explication :**

Une methode qui crée et ajoute une ligne

#### 3. ajoutStat :

```
static void ajoutStat(ArrayList<ArrayList<Ligne>> Lignes)
```



### **Explication :**

Une methode qui crée et ajoute une station

#### 4.MOYt :

```
static MoyTransport.MoyType MOYt(String s)
```

### **Explication :**

Une methode qui retourne le moyen de transport a partir d'une abréviation

#### 5.getTypeFromID :

```
static String getTypeFromID(String s)
```

### **Explication :**

Une methode qui retourne une abréviation d'un moyen de transport

#### 6.getTransFromId :

```
static MoyTransport.MoyType getTransFromId(String id)
```

### **Explication :**

Une methode qui retourne le type a partir de l'id

#### 7.SupLigne :

```
static void SupLigne(ArrayList<ArrayList<Ligne>> Lignes)
```

### **Explication :**

Une methode qui supprime une ligne

#### 8.SupStat :

```
static void SupStat(ArrayList<ArrayList<Ligne>> Lignes)
```

### **Explication :**

Une méthode qui supprime une station

#### 9.statprinc:

```
static void statPrinc(ArrayList<ArrayList<Ligne>> Lignes, Station S1)
```

### **Explication :**

Une méthode qui détermine les station condescendante et principale et les mets dans un arraylist

#### 10 .StatEQcord :

```
static String StatEQcord(ArrayList<ArrayList<Ligne>> Lignes, Cord c)
```

### **Explication :**

Une méthode qui teste si les coordonnées existentent déjà , retourne le nom de la station qui lui est egale, pour l'affecter a une nouvelle station

#### 11. addNav :

```
static void addNav(ArrayList<ArrayList<Ligne>> Lignes)
```

### **Explication :**

Une methode qui ajoute une navette

#### 12.SupNav :

```
static void SupNav(ArrayList<ArrayList<Ligne>> Lignes)
```

### **Explication :**

Une methode qui supprime une navette

#### 13 .ModifNavette :

```
static void ModifNavette(ArrayList<ArrayList<Ligne>> Lignes)
```

### **Explication :**

Une methode qui modifie une navette

#### 14.ModLigne :

```
static void ModLigne(ArrayList<ArrayList<Ligne>> Lignes)
```

### **Explication :**

Une méthode qui modifie une ligne

15. modifStat :

```
static void ModifStat(ArrayList<ArrayList<Ligne>> Lignes)
```

### **Explication :**

Une methode qui modifie une station

14. PRINC :

```
static void PRINC(ArrayList<ArrayList<Ligne>> Lignes)
```

### **Explication :**

Une méthode qui détermine les station principal et condescendance , et les met dans un array list (séparément)

## **Partie II :Requetes d'interrogation**

Qst1 :Affichage de toutes les lignes de transport

```
static void qst1 (ArrayList<ArrayList<Ligne>> Lignes)
```

Qst2 : affichage le chemin d'une ligne donnée

```
static void qst2(ArrayList<ArrayList<Ligne>> Lignes)
```

Qst3 : chercher les lignes qui passent par une station

```
static void qst3(ArrayList<ArrayList<Ligne>> Lignes)
```

Qst4: affichage de station de correspondances

```
static void qst4(ArrayList<ArrayList<Ligne>> Lignes)
```

Qst5: affichage de la ligne empruntée par le plus grand nombre de voyageurs

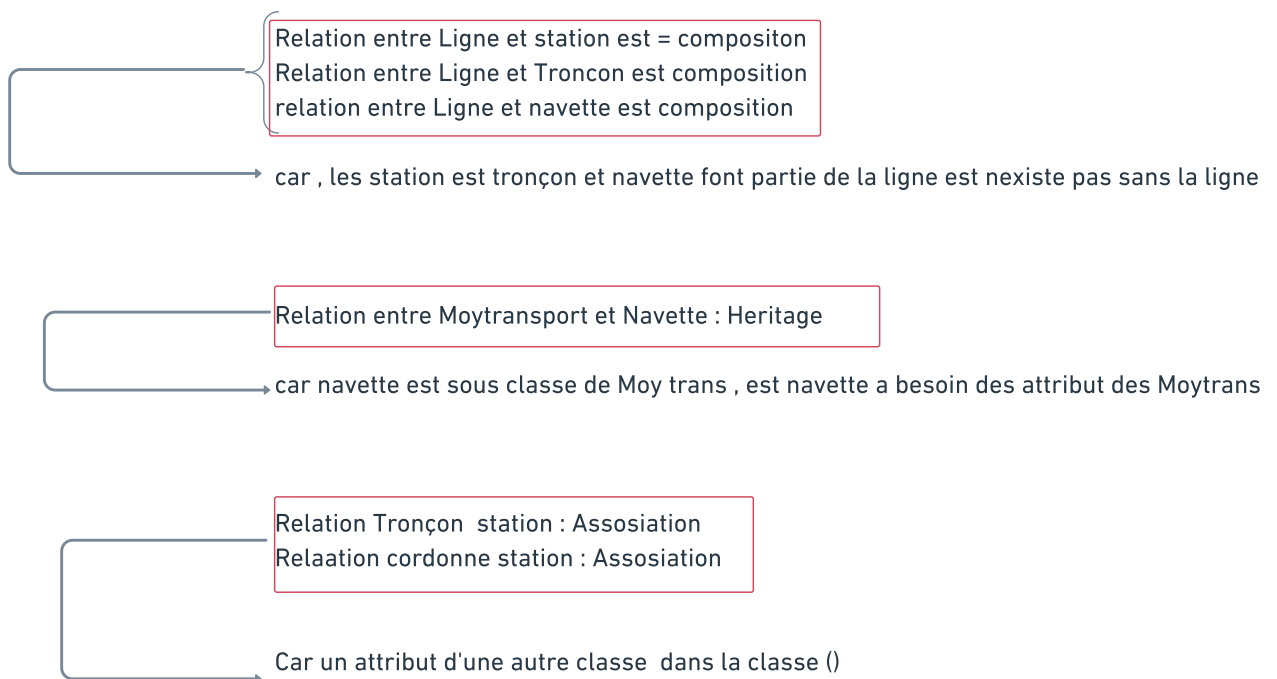
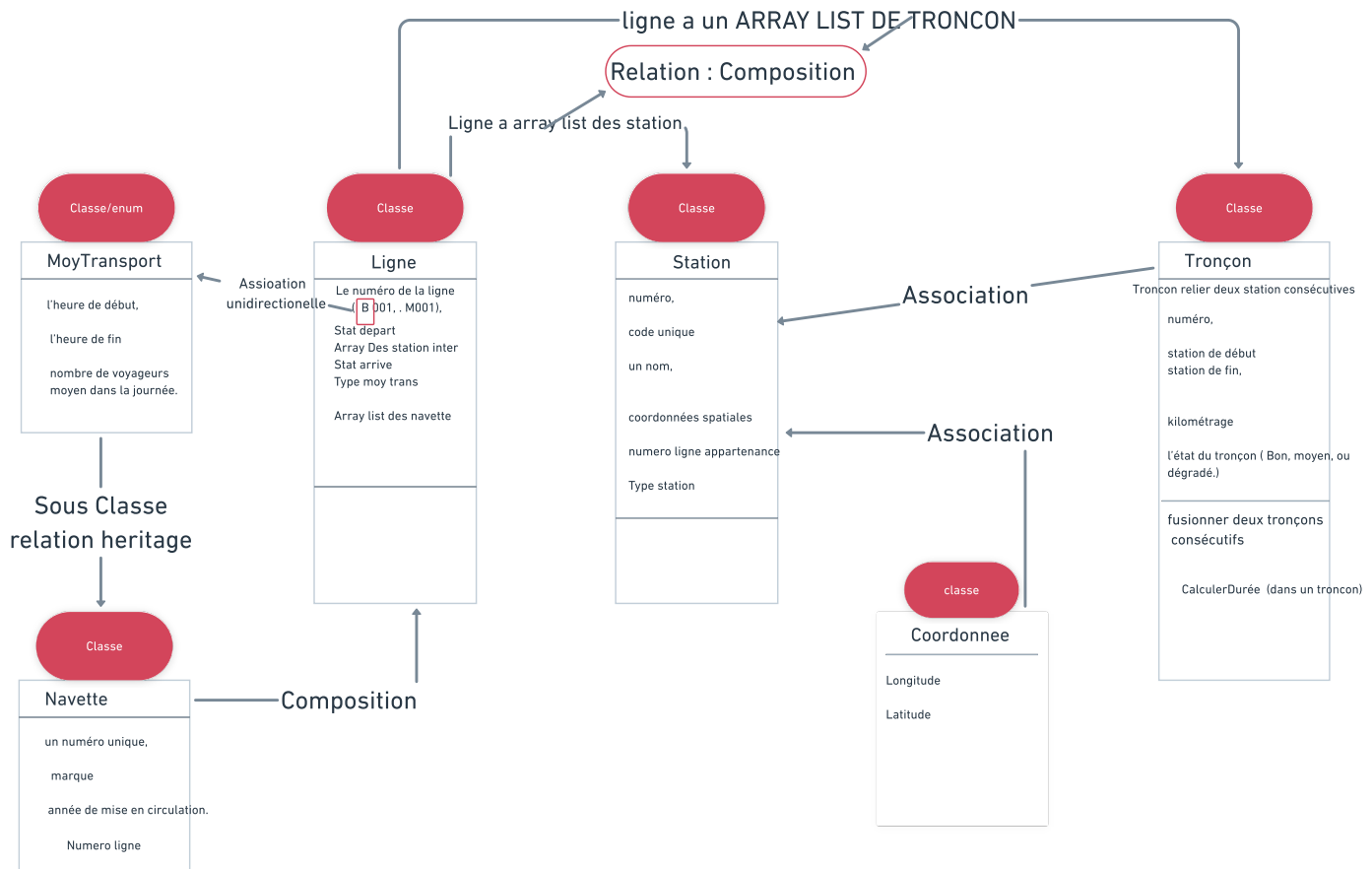
```
static void qst5(ArrayList<ArrayList<Ligne>> Lignes)
```

Qst6: calculer la durée estimé pour aller d'un point A à un point B

```
static void qst6(ArrayList<ArrayList<Ligne>> Lignes)
```

Qst 7: Construction de chemin pour aller du point A au point B

```
static void qst7(ArrayList<ArrayList<Ligne>> Lignes)
```



### 3) Jeu d'essay :

#### -Menu principal :

```
Remplissage Auto ██████████  NOW LOADING

✎ Remplissage automatique fait ✎

      ||
      -----
    | Projet TP de P00 |
      -----

    | Que voulez vous faire ? |

-----
|                               |
|   1. Faire une requete de mise a jour   |
|   2. Faire une Requêtes d'interrogation |
|   3. Quitter                             |
|                               |
-----

Veuillez introduire un numero:
```

```
Les operations de mise a jour possible :

-----
| 1. Modification                       |
|                                     |
| 2. Ajout                             |
|                                     |
| 3. Suppression                       |
|                                     |
| 4.-- Revenir au menu précédent --   |
|                                     |
-----

Veuillez introduire un numero:
|
```

```
vous voulez faire l'ajout de :
-----
| 1. Ligne                               |
|                                       |
| 2. Station                           |
|                                       |
| 3. Tronçon,                          |
|                                       |
| 4. -- Revenir au menu précédent --  |
|                                       |
-----

Veuillez introduire un numero:
```

## Requete Ajout Ligne :

```
Ajout d'une ligne :

- Introduction Du ID unique de la ligne -

Precisez le Type de la Ligne : metro
Insere le numero de la ligne :12

- Introduction Des Station -

Saisissez ses Coordonee GPS : 2.3 6.5
null
Saisissez le nom de la station : alger
- Voulez Vous Rajouter une autre station : (Y/N) -
```

Si on introduit une station deja introduite dans la meme ligne , on re demnde de re introduire une nouvelle station :

```

y
Saisissez ses Coordonee GPS : 2.3 6.5
null
Saisissez le nom de la station : alger
    !_! Station existe Deja !! introduisez une autre !_!
Saisissez ses Coordonee GPS : 1.6 9.3
null
Saisissez le nom de la station : blida
    Voulez Vous Rajouter une autre station : (Y/N)

```

```

y
Saisissez ses Coordonee GPS : 15.2 6.9
null
Saisissez le nom de la station : wahran
    - Voulez Vous Rajouter une autre station : (Y/N) -
n
    -= Ligne 'M012' Cree ✓  -=

```

Et on propose si il veut introduire une navette :

```

    - Voulez vous rajouter des navette a cette ligne ? (Y/N) -
y
Les Navette Deja Introduite : []
intrduire le Numero de la navette : 15
la marque : chevrolet
introduire année de mise en circulation : 2001
    - Voulez vous rajouter des navette a cette ligne ? (Y/N) -

```

## Requete Suppression Navette :



Veuillez introduire un numero:

3

vous voulez faire la suppression de :

```
-----  
| 1. Ligne |  
|         |  
| 2. Station |  
|         |  
| 3. Navette |  
|         |  
| 4. -- Revenir au menu précédent -- |  
-----
```

Veuillez introduire un numero: 3

Suppresion d'une navette :

Affichage De Toutes Les Navette

-----  
=■| Les Navette Du Type ' BUS ' :

|| Ligne B007 -> [ Aucune navette ] || Ligne B019 -> [5, 6]

=■| Les Navette Du Type ' TRAIN ' :

|| Ligne T009 -> [ Aucune navette ] || Ligne T003 -> [5, 6]

=■| Les Navette Du Type ' TRAMWAY ' :

|| Ligne TR014 -> [ Aucune navette ] || Ligne TR016 -> [3, 4]

=■| Les Navette Du Type ' METRO ' :

|| Ligne M002 -> [1, 2] || Ligne M004 -> [ Aucune navette ] || Ligne M012 -> [15]

-----  
Veuillez Choisir une Ligne :

-----  
Veuillez Choisir une Ligne : **b019**

-----  
Les Navette De Cette Ligne : [5, 6]

-----  
Veuillez Choisir une Navette : **6**

-----  
Les Navette De Cette Ligne : [5]

-----  
Navettes Supprimer ✓

-----  
=■■■| Les Navette Du Type ' BUS ' :

|| Ligne B007 -> [ Aucune navette ] || Ligne B019 -> [5]

=■■■| Les Navette Du Type ' TRAIN ' :

|| Ligne T009 -> [ Aucune navette ] || Ligne T003 -> [5, 6]

=■■■| Les Navette Du Type ' TRAMWAY ' :

|| Ligne TR014 -> [ Aucune navette ] || Ligne TR016 -> [3, 4]

=■■■| Les Navette Du Type ' METRO ' :

|| Ligne M002 -> [1, 2] || Ligne M004 -> [ Aucune navette ] || Ligne M012 -> [15]

-----

**Requete Modification Station :**

Veuillez introduire un numero: 2

Modication d'une station :

-----  
LIGNE 'B007' : Primrose Hill Road (1) - Chalk Farm (2) - Gibson (3) - Camden Town (4) - Camden Road (5) - Royal College (6) - Randells Road (7) - Kings Cr  
LIGNE 'B019' : London Bridge Bus (11) - Monument (12) - King William Street (13) - Bank Station (14) - Canonbury (15) - Bishopsgate (16) - Stop A (17) - A  
LIGNE 'T009' : Euston Square (21) - Kings Cross St Pancras (8) - Highbury et Islington (23) - Drayton Park (24) - Arsenal (25) - Manor House (26) - Seven S  
LIGNE 'T003' : Kentish Town West (31) - Camden Town (4) - Caledonian Road (33) - Highbury et Islington (23) - Canonbury (35) - Dalston Kingsland (36) - Hag  
LIGNE 'TR014' : Leicester Square (41) - Covent Garden (42) - Holborn (43) - Russell Square (44) - Kings Cross St Pancras (8) - Caledonian Road (33) - Arser  
LIGNE 'TR016' : Edgware (51) - Colindale (52) - Golders Green (53) - Hendon Central (54) - Camden Town (4) - Mornington Crescent (56) - Warren Street (57)  
LIGNE 'M002' : Aldgate (18) - Liverpool Street (19) - Moorgate (20) - Barbican (64) - Kings Cross St Pancras (65) - Euston Square (66) - Great Portland Str  
LIGNE 'M004' : Stratford (71) - Mile End Station (72) - Bethnal Green (73) - Liverpool Street (19) - Bank Station (14) - St Pauls (76) - Chancery Lane (77)  
LIGNE 'M012' : alger (81) - blida (83) - wahran (84) -  
-----

Veuillez Choisir une Ligne :

-----  
Veuillez Choisir une Ligne : t009

Choisissez une station (par nom) : arsenal

4

1 - Modification Du Nom De La Station

2 - Modification Des Coordonee

1

Nouveau Nom de la Station : ALGER

middle

ALGER

-----  
LIGNE 'T009' :

Euston Square	(21)->	Kings Cross St Pancras (8)->	Highbury et Islington (23)->	Drayton Park	(24)->
ALGER	(25)->	Manor House	(26)->	Seven Sisters	(27)->
Blackhorse Road	(29)->	Walthamstow Central	(29)	Tottenham Hale	(28)->

-----

Station Modifier ✓

Requete D'Interrogation :

```

      ★ Les requetes d'interrogation possible
-----
| 1.      Affichage de toutes les lignes de transport      |
| 2.      Affichage du chemin sur une ligne donnée         |
| 3.      Chercher toutes les lignes qui passent          |
|           une station donnée                             |
| 4.      Affichage des stations de correspondance         |
| 5.      Affichage La ligne empruntée par le plus        |
|           grand nombre de voyageurs                     |
| 6.      Affichage La durée estimée pour aller d'un     |
|           point A à un point B                         |
| 7.      Construction de chemin pour aller d'un point   |
|           A à un point B                                |
| 8.      -- Revenir au menu précédent --                 |
|-----
introduire un numero: |

```

## Requette Cheminn Ligne Donnee :

```

Veuillez introduire un numero: 2
- Affichage de toutes les lignes de transport -
-----
LIGNE 'B007' : BUS | Station Depart -> Primrose Hill Road | Station Arrivee -> Caledonian Road
LIGNE 'B019' : BUS | Station Depart -> London Bridge Bus | Station Arrivee -> Moorgate
LIGNE 'T009' : TRAIN | Station Depart -> Euston Square | Station Arrivee -> Walthamstow Central
LIGNE 'T003' : TRAIN | Station Depart -> Kentish Town West | Station Arrivee -> Liverpool Street
LIGNE 'TR014' : TRAMWAY | Station Depart -> Leicester Square | Station Arrivee -> Bounds Green
LIGNE 'TR016' : TRAMWAY | Station Depart -> Edgware | Station Arrivee -> Leicester Square
LIGNE 'M002' : METRO | Station Depart -> Aldgate | Station Arrivee -> St Johns Wood
LIGNE 'M004' : METRO | Station Depart -> Stratford | Station Arrivee -> Oxford Circus Underground Station
LIGNE 'M012' : METRO | Station Depart -> alger | Station Arrivee -> wahran
-----
- Affichage du chemin sur une ligne donnée -
Veuillez Introduire Une Ligne

```

On remarque la ligne Qu'on a ajouter au debut est bien sauvegarder

```

- Affichage du chemin sur une ligne donnée -
Veuillez Introduire Une Ligne
m004
LIGNE 'M004' :
    Stratford          (71)->    Mile End Station    (72)->    Bethnal Green    (73)->    Liverpool Street    (19)->
    Bank Station      (14)->    St Pauls        (76)->    Chancery Lane    (77)->    Holborn          (43)->
    Tottenham Court Road (79)-> Oxford Circus Underground Station (79)

Les Station Principale : - Liverpool Street - Liverpool Street - Bank Station - Holborn

```

## Requette Afiichage des station de correspondance:

```

Veuillez introduire un numero: 4

- Affichage Des Station De Correspondance -

Camden Town          Kings Cross St Pancras    Bank Station            Aldgate                Liverpool Street
Moorgate             Highbury et Islington    ALGER                   Manor House
Caledonian Road      Holborn

```

On remarque que : ALGER est correspondance , mais ca represente la station Arsenal Qu'on a change le nom au debut

## Affichage de toutes les lignes de transport :

```

Veuillez introduire un numero: 1

- Affichage de toutes les lignes de transport -
-----
LIGNE 'B007' : BUS | Station Depart -> Primrose Hill Road | Station Arrivee -> Caledonian Road
LIGNE 'B019' : BUS | Station Depart -> London Bridge Bus | Station Arrivee -> Moorgate
LIGNE 'T009' : TRAIN | Station Depart -> Euston Square | Station Arrivee -> Walthamstow Central
LIGNE 'T003' : TRAIN | Station Depart -> Kentish Town West | Station Arrivee -> Liverpool Street
LIGNE 'TR014' : TRAMWAY | Station Depart -> Leicester Square | Station Arrivee -> Bounds Green
LIGNE 'TR016' : TRAMWAY | Station Depart -> Edgware | Station Arrivee -> Leicester Square
LIGNE 'M002' : METRO | Station Depart -> Aldgate | Station Arrivee -> St Johns Wood
LIGNE 'M004' : METRO | Station Depart -> Stratford | Station Arrivee -> Oxford Circus Underground Station
LIGNE 'M012' : METRO | Station Depart -> alger | Station Arrivee -> wahran
-----

```

Le reste de L'execution sera dans la video de presentation , Merci

**Que doit-on modifier dans la conception des classes si l'on doit considérer qu'il existe des lignes permanentes (c'est à dire des lignes en service même les jours de week-end et les jours fériés) en précisant les horaires de service pour ces jours spécifiques ? Ces horaires peuvent changer dans l'année.?**

On pense que afin de faire une class type de Ligne ,  
et faire deux classe fille qui auront cahqun le type enum spécifier au journee ferier ou non , cahue sous classe aura temps different pour chaque moyen