

République Algérienne Démocratique et Populaire Ministère de
L'enseignement Supérieur et de la Recherche Scientifique
Université des Sciences et de la Technologie Houari Boumediene
Faculté d'Electronique et d'Informatique



Rapport Mini projet en Programmation orientée objet

RÉALISÉ PAR :

LOUNAS Katia

HAMI Islam

Table des matières

I-Introduction.....	2
II- Objectif.....	2
III- Analyse et Conception	3
Partie I : Description des classes	3
Classes élémentaires :	3
1.Enumération MoyTransport :	3
2. Classe Ligne :	3
3. Classe Station :	4
4. Classe Tronçon :	5
5. Classe Navette :	6
Classes Supplémentaires :	6
6.Enumération EtatTronçon :	6
7.Classe Ligne_List :	6
Classes de l'interface Graphique :	7
IV-Jeu d'essai	8
Ajout d'une ligne :	8
Ajout d'une station :	9
Suppression d'une Ligne :	9
Affichage de toutes les lignes :	10
Affichage des stations d'une ligne sur un axe :	10
Affichage des lignes qui passe par une station :	11
Affichage des stations de correspondance :	11
Affichage de la ligne empruntée par le plus grand nombre de voyageurs :	12
Calcule de la durée :	12
Affichage menu sous console :	13
Affichage de l'interface Graphique :	14
V- Conclusion.....	16

I-Introduction

Depuis la nuit des temps, l'homme entretient avec la technologie un rapport ambigu. Fasciné par le progrès et la modernité qu'elle symbolise, auxquels il voue une passion obsessive à évoluer, à ce propos il a conçu des outils plus adaptés afin de lui faciliter la tâche.

Nous allons donc nous intéresser à la conception et réalisation d'une petite application appelée « MySmartCity » qui fera l'objet de la gestion du réseau de transport urbain dans une ville intelligente. Pour s'en faire on utilise la POO avec le langage Java.

La Programmation Orientée Objet, une méthode de programmation informatique de plus en plus plébiscitée, que ce soit dans le développement logiciel ou la data science. Organisée autour des objets, ou données, la POO offre de nombreux avantages.

II- Objectif

- Appliquer les principes de la POO afin d'implémenter une application en Java.
- Réaliser des différents traitements sur les classes.
- Utiliser le principe d'encapsulation.
- Manipuler les différentes bibliothèques afin d'utiliser ses méthodes.
- Concevoir un menu à choix multiples pour l'exécution des requêtes.

III- Analyse et Conception

Partie I : Description des classes

Classes élémentaires :

1. Enumération MoyTransport :

Identificateur : **B** (bus), **T**(tramway), **TR** (train), **M**(métro).

Attribut : int heure-de-début, heure-de-fin, nbr-de-voyageurs.

Constructeur : MoyTransport (int heur_debut, int heure_fin, int nbr_moyen_voyageurs)

Méthodes :

- Public String Mtransport () ;

2. Classe Ligne :

Attribut : . String numero ;

- MoyTransport M ;
- Station station_de_depart, station_d_arrivee ;
- Int nbr ; //nombre de station
- ArrayList < Station > ST ;
- ArrayList <Tronçon> TR ;
- ArrayList <Navette> NA ;

Constructeur : public Ligne () {} //constructeur sans paramètre.

Public Ligne (String numero, ArrayList < Station > ST , ArrayList <Tronçon> TR
ArrayList <Navette> NA)

Méthodes :

- Public String toString() ;
- Public void afficher () ;
- Public Ligne Saisie() ;
- Public ArrayList<Ligne> Creation() ;
- Public int Recherche_NA(int i) //Recherche d'une navette dans la collection navette
- Public void Modifier_NA(int i) ; //Modification d'une navette

- Public void Supprimer_NA(int i) ; // suppression d'une navette
- Public void Ajout_NA(int i) ; // Ajout d'une navette
- Public void Modifier_TR(int i) ; //Modification d'un tronçon
- Public void Supprimer_TR(int i) ; // suppression d'un Tronçon
- Public void Ajout_TR(int i) ; // Ajout d'un tronçon
- Public int Recherche_ST(Station S) ; // Recherche d'une station
- Public void Modifier_ST(int i) ; //Modification d'une station
- Public void Supprimer_ST(int i) ; // suppression d'une station
- Public void Ajout_ST(int i) ; // Ajout d'une station

Accesseurs : Public String getNumero () ;

- Public Station getStation_d_arrivee () ;
- Public Station getStation_de_depart () ;
- Public int getNbr () ;
- Public ArrayList <Station> getST () ;
- Public ArrayList <Navette> getNA () ;
- Public ArrayList <Tronçon > getTR () ;
- Public void setNumero(String numero) ;
- Public void setNbr (int nbr) ;

3. Classe Station :

Attribut : . Int numero ;

- String nom, type ;

Constructeur : public Station () {} ;

Public Station (int numero, String nom, String type) ;

Méthodes : . Public Station Saisie() ;

- Public ArrayList<Station> Creation (int nbr) ;
- Public String toString() ;
- Public void afficher () ;

Accesseurs :

- Public String getNom () ;
- Public String getType () ;
- Public int getNumero (int numero) ;
- Public void setNumero (int numero) ;
- Public void setNom (String nom) ;
- Public void setType (String type) ;

4. Classe Tronçon :

Attribut : . Int numero ;

- Station station_de_debut ;
- Station station_de_fin ;
- Double kilometrage ;
- EtatTronçon Etat_troncon ;

Constructeur : public Tronçon () {} ;

Public Tronçon (int numero, Station station_de_debut, Station station_de_fin, Double kilometrage, EtatTronçon Etat_troncon) ;

Méthodes : . Public Station Saisie () ;

- Public void creation () ;
- Public String toString () ;
- Public void afficher () ;

Accesseurs : . public int getNumero () ;

- Public void setNumero (int numero) ;
- Public Station getStation_de_debut () ;
- Public void setStation_de_debut (Station station_dr_debut) ;
- Public Station getStation_de_fin () ;
- Public void setStation_de_fin (Station station_dr_debut) ;
- Public double getKilometrage () ;
- Public void setKilometrage (double kilometrage) ;

- Public EtatTronçon getEtat_Troncon () ;
- Public void setEtat_de_Troncon (EtatTronçon etat_de_troncon) ;

5. Classe Navette :

Attribut : . Int num_unique ;

- String marque ;
- Int annee ;

Constructeur : public Navette () {} ;

Public Navette (Int num_unique, String marque, Int annee) ;

Méthodes : . Public Station Saisie () ;

- Public String toString () ;
- Public void afficher () ;
- Public ArrayList<Station> Creation () ;

Accesseurs : . public int getNum_unique () ;

- Public void setNum_unique (int num_unique) ;
- Public String getMarque () ;
- Public void setMarque (String marque) ;
- Public int getAnnée () ;
- Public void setAnnée (int annee) ;

Classes Supplémentaires :

6.Énumération EtatTronçon :

Identificateur : Bon, moyen, dégradé.

7.Classe Ligne_List :

Rôle : c'est la classe qui contient toutes les méthodes qui manipule la collection des lignes.

Attribut : ArrayListe <Ligne> LN ;

Constructeur : public Ligne_List (){}

public Ligne_List (ArrayListe <Ligne>)

Méthodes : . public int Recherche_LN(String numero) //recherche d'une ligne dans la collection

- Public void Modifier_LN(int i) //modification d'une ligne
- Public void Ajout_LN(int i) //Ajout d'une ligne
- Public void Affichage() //Affichage des ligne de la collection
- Public void affiche_Axe(String num) //Affichage d'une ligne avec ses stations ..
- Public void Station_Correspondance()
- Public void Grand_nb_Voyageur()
- Public void Petit_nb_Voyageur()
- Public String Calcul_Duree()
- Public void Remplissage_auto()

Accesseurs: public ArrayList<Ligne> getN()

Classes de l'interface Graphique :

L'interface n'est pas a 100% fonctionnelle, a cote y l'affichage dans la console qui est fonctionnel.

Pour l'interface on a utilisé la bibliothèque Swing on a construit des différentes classes avec des JFrame pour gérer les évènements sur les boutons on a utilisé des classes anonymes action listnner.

- Public Classe Window
- Public Classe WindowPrincipale
- Public Classe MiseAJourBoutton
- Public Classe RechercheBoutton
- Public Classe RechercheBoutton2
- Public Classe Ajout
- Public Classe Ajout_Ligne
- Public Classe Ajout_Station
- Public Classe Ajout_Navette
- Public Classe Ajout_Troncon
- Public Classe Duree

IV-Jeu d'essai

Pour les requêtes de modification c'est la même chose pour chacun de ligne, station, navette, tronçon.
Dans notre jeu d'essai on va aborder juste quelques cas

Ajout d'une ligne :

Ligne

Numéro	005	Nom station arrivée	Tizi ousou
Moyen de transport	Bus	Num station arrivée	163
Nbr de station	2	Type station arrivée	Secondaire
Nom Station Départ	Alger	Num navette	12
Num station départ	107	Marque navette	BMW
Type station départ	Principale	Année fabrication	2019
Num tronçon	005	Kilométrage	12
Etat tronçon	Bon	/	/

Capture d'Ecran :

```
saisie des donnees de la ligne:
Donnez le numero de la ligne:
005
Donnez le moyen de transport:
En introduisant: B ( bus) , TR( Train) ,T (Tramway) , M (Metro)
B
Donnez le nombre de Station
2
Donnez le nombre de Navette
1
Donnez le numero de la Navette:
12
Donnez la marque de la Navette :
BMW
Donnez l'année de fabrication de la Navette :
2019
Donnez le numero de la Station :
107
Donnez le nom de la Station :
Alger
Donnez le type de la station :
Principale
```

```
Donnez le numero de la Station :
163
Donnez le nom de la Station :
Tizi ousou
Donnez le type de la station :
Secondaire
Donnez le numero du Tronçon:
005
Donnez la longueur du Tronçon en kilometre :
12
Donnez l'etat du Tronçon :( Bon ,Moyen, Dégradé)
Bon
--l'Ajout a été fait avec succes !!--
```

Ajout d'une station :

Station :

Numéro Ligne	TR007	Type de station	Principale
position	2	Num station	123
Longueur de tronçon	1234	Nom de station	ABC
Etat tronçon	Bon	Num tronçon	123

Capture d'Ecran

```

Saisissez Ligne dans laquelle la station sera ajouter:
TR007
a quel position parmi les 8 Station vous le vous ajouter:
2
Donnez le numero de la Station :
123
Donnez le nom de la Station :
ABC
Donnez le type de la station :
Principale
Donnez le numero du Tronçon:|
123
Donnez la longueur du Tronçon en kilometre :
1234
Donnez l'etat du Tronçon :( Bon ,Moyen, Dégradé)
Bon
--l'Ajout a été fait avec succes !!--

```

Suppression d'une Ligne :

Ligne :TR007 qui se trouve dans la position 1

Capture d'Ecran

```

1- Ajout
2-Modification:
3-Suppression
3
Quel ligne voulez vous supprimer : (4 lignes deja introduites )
1
----LA LIGNE A ETE SUPPRIMER---

```

Affichage de toutes les lignes :

```

-----LES LIGNES SONT:-----
ligne numero  TR007
moyen de transport Train
station :
Depart  Gare de l'est numero 109 de type principale
-----
Arrivee  Lille Europe numero 814 de type secondaire
-----

-----

ligne numero  M001
moyen de transport Metro
station :
Depart  Alexandre Dumas numero 104 de type principale
-----
Arrivee  Jasmin numero 801 de type secondaire
-----

```

Affichage des stations d'une ligne sur un axe :

Ligne : T004

```

Donnez la ligne à afficher:
Ligne deja existante : T004,M001,B005,TR007
T004
Alexandre Dumas(principale)-----Argentine-----Eglise de Pantin-----Courrounes-----Croix de Chavaux(principale)-----Jasmin-----

```

Affichage des lignes qui passe par une station :

Station : 109

```
Donnez le numero de la Station:
Station deja existante : 504,704,104,801,,647,814,109
109
ligne numero   TR007
moyen de transport Train
station :
Depart  Gare de l'est numero 109 de type principale
-----
Arrivee  Lille Europe numero 814 de type secondaire
-----
```

Affichage des stations de correspondance :

```
Les stations de correspondance sont:

Gare de l'est numero 109 de type principale
-----
Gare Montparnasse numero 602 de type principale
-----
Alexandre Dumas numero 104 de type principale
-----
Croix de Chavaux numero 647 de type principale
-----
Alexandre Dumas numero 104 de type principale
-----
Croix de Chavaux numero 647 de type principale
-----
Portes des Lilas numero 504 de type principale
-----
```

Affichage de la ligne empruntée par le plus grand nombre de voyageurs :

```
La ligne avec le plus grand nombre de voyageur est:

ligne numero  TR007
moyen de transport Train
station :
Depart  Gare de l'est numero 109 de type principale
-----
Arrivee  Lille Europe numero 814 de type secondaire
-----
```

```
-----
La ligne avec le plus petit nombre de voyageur est :

ligne numero  T004
moyen de transport Tramway
station :
Depart  Alexandre Dumas numero 104 de type principale
-----
Arrivee  Croix de Chavaux numero 647 de type principale
-----
```

Calcule de la durée :

Ligne :T004

Position A : 13

Position B : 5

```
Donnez la ligne:
T004
Donnez les deux positions des 2 stations A et B parmi les 6 Station
13
5

la Durée est: la durée est: 22h10m32s
```

Affichage menu sous console :

```
-----
                        BIEVENUE
-----

-----

REQUETE SUR LES DONNEES:

-----
Choisissez la requete:

1- Requete de Mise à jour:

2-Requete d'interrogation:
2
1- Affichage de toutes les lignes .
2- Affichage du chemin sur une ligne donnée
3- Chercher toutes les lignes qui passent par une station donnée
4- Affichage des stations de correspondance
5- La ligne empruntée par le plus grand nombre de voyageurs et par le minimum de voyageurs
6-La durée estimée pour aller d'un point A à un point B,
```

```
REQUETE SUR LES DONNEES:

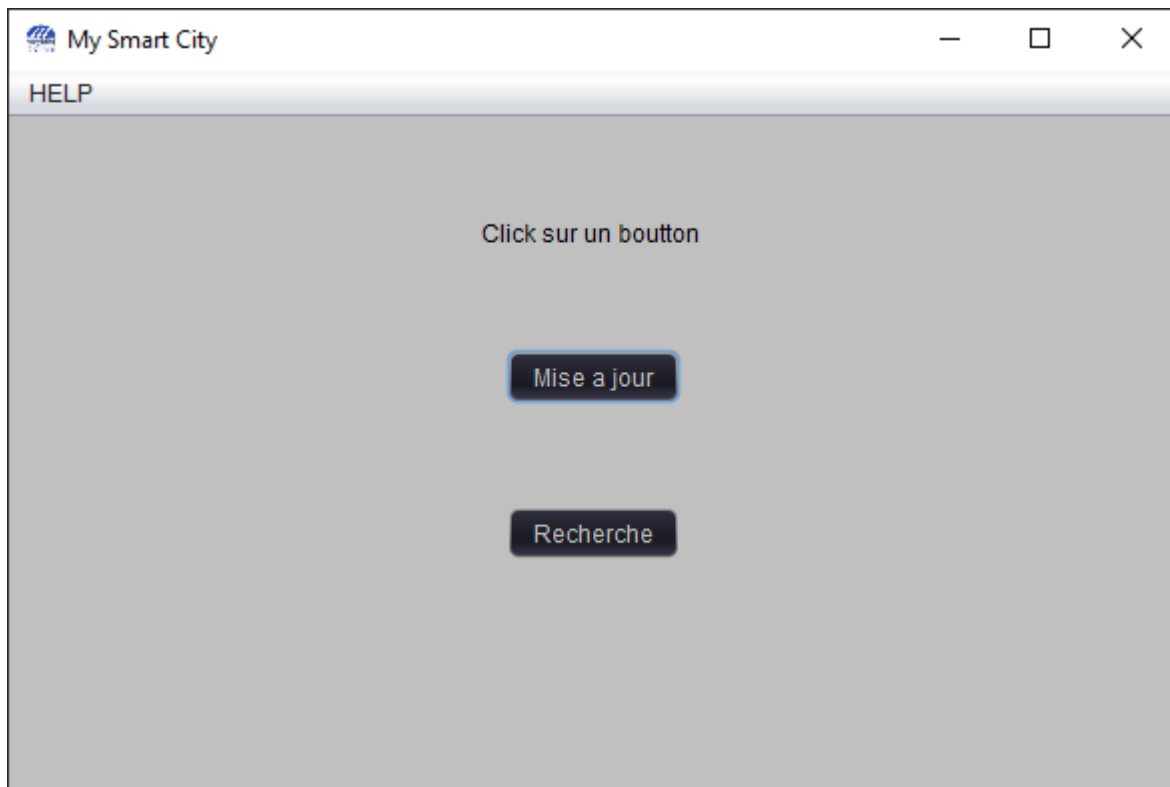
-----

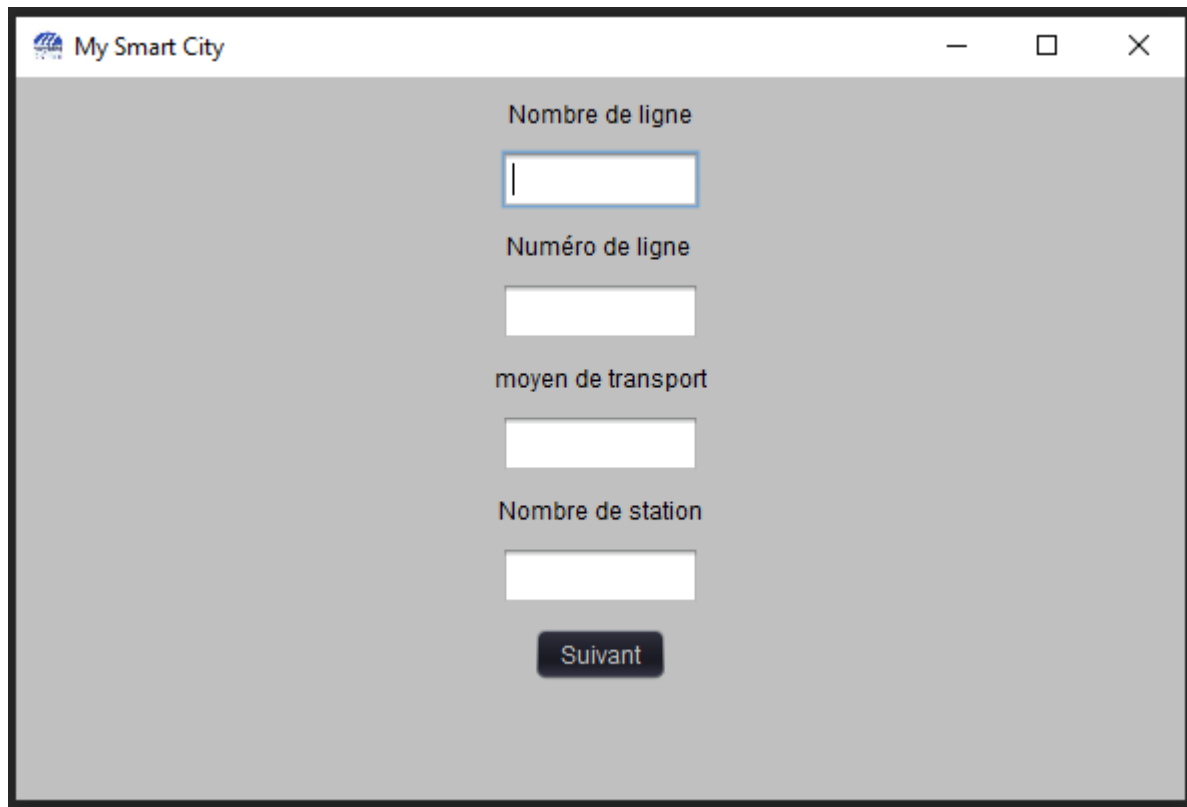
Choisissez la requete:

1- Requete de Mise à jour:

2-Requete d'interrogation:
1
Sur quel partie voulez vous faire la mise à jour:

1- Ligne
2-Station
3-Tronçon
4-Navette
```

Affichage de l'interface Graphique :



The screenshot shows a web application window titled "My Smart City". The main content area is a light gray rectangle. Inside, there are five labels with corresponding input fields, all centered vertically and horizontally. The labels are "Nombre de ligne", "Numéro de ligne", "moyen de transport", and "Nombre de station". The input fields are white rectangles. The first field for "Nombre de ligne" contains a single vertical bar "|". Below these fields is a dark gray button with the white text "Suivant".

Nombre de ligne

Numéro de ligne

moyen de transport

Nombre de station

Suivant



The screenshot shows the same "My Smart City" application window, but at a different step. The labels are "Posistion point A" and "Posistion point B". The input fields are white rectangles. The first field for "Posistion point A" contains the number "2". The second field for "Posistion point B" contains the number "12". Below these fields, the text "la durée est: 13h21m20s" is displayed. At the bottom, there are two buttons: a dark gray button with the white text "Afficher" and a light gray button with the dark gray text "Retour".

Posistion point A

Posistion point B

la durée est: 13h21m20s

Afficher

Retour

V- Conclusion

S'il y a bien une chose dont notre époque a l'immense chance de jouir, ce n'est que l'outil informatique. On constate que les multiples propriétés qui ont poussé l'homme à la recherche, on prit un temps considérable, sont résolus au moins de temps.

A travers ce projet on a pu aborde quelques exemples du pouvoir de la Programmation Orientée Objet, Un outil qui a réduire la complexité de la conception des applications avec la méthode procédurale.