

Ruben BLANCUZZI
Noé DOUDOU
Maxime ERNANDEZ
Paul HOPSORE
Marek SEDLACEK



Rapport Projet Java

2e Rendu : Implémentation Java en mode texte

Introduction et objectifs :

Dans ce projet, nous nous intéressons à un système de gestion et de tri des déchets. Après avoir réalisé la modélisation UML des acteurs du système, nous passons maintenant à l'implémentation en Java de ses principales fonctionnalités en mode texte. L'objectif de ce rendu est de traduire la modélisation en un code fonctionnel, structuré en classes et packages, et de valider son bon fonctionnement à l'aide de tests unitaires. Cette implémentation servira de fondation pour l'ajout d'une interface graphique lors de la prochaine étape du projet

Organisation du travail :

Pour organiser au mieux notre travail, nous avons utilisé un serveur discord dédié, sur lequel chacun partageait ses avancées, et nous avons prévu des réunions de manière régulière afin de faire le point sur l'avancée du rendu. C'est un outil qui s'est révélé très efficace pour travailler en groupe, notamment grâce aux appels et à la fonction de partage d'écran.

Pour l'implémentation en Java, nous avons utilisé CodeTogether afin de pouvoir travailler simultanément sur le projet. Cet outil nous a permis de coder en collaboration en temps réel, de partager nos écrans et de commenter directement les parties du code en cours de développement. Grâce à cette approche, nous avons pu identifier et corriger plus rapidement les erreurs et assurer une meilleure cohérence dans l'implémentation des différentes classes. L'utilisation conjointe de CodeTogether et de Discord a facilité la communication et la répartition des tâches, optimisant ainsi notre efficacité tout au long du projet.

Scénario typique du Gestion de Tri sélectif

(Main)

On crée un Centre de tri nommé centreA. On lui ajoute 4 poubelles intelligentes.

```
8• public static void main(String[] args) {
9
10     //On crée un centre de tri
11     CentreDeTri centreA=new CentreDeTri("CentreA","Paris");
12
13     //On lui ajoute des poubelles
14     centreA.ajouter("3 avenue des Champs-Élysées",0);
15     centreA.ajouter("13 rue de general de Gaulle",1);
16     centreA.ajouter("10 rue Bonaparte",2);
17     centreA.ajouter("23 rue Paul Vaillant Couturier",3);
18 }
```

Affichage le centreA pour vérifier que les poubelles intelligentes ont bien été ajoutées.

```
25 //On affiche les infos du centre de tri pour vérifier que les poubelles ont bien été ajoutées
26 System.out.println(centreA.toString()+"\n \n");
```

```
( Id du Centre de Tri : 1, Nom du Centre de Tri : CentreA, Adresse du centre de Tri : Paris,
Liste des poubelles :
/ Id de la poubelle : 1, Emplacement poubelle : 3 avenue des Champs-Élysées, Type poubelle: 0, Poubelle pleine ? : false,
Liste des dépôts : VIDE./ ,
/ Id de la poubelle : 2, Emplacement poubelle : 13 rue de general de Gaulle, Type poubelle: 1, Poubelle pleine ? : false,
Liste des dépôts : VIDE./ ,
/ Id de la poubelle : 3, Emplacement poubelle : 10 rue Bonaparte, Type poubelle: 2, Poubelle pleine ? : false,
Liste des dépôts : VIDE./ ,
/ Id de la poubelle : 4, Emplacement poubelle : 23 rue Paul Vaillant Couturier, Type poubelle: 3, Poubelle pleine ? : false,
Liste des dépôts : VIDE./ ,
Historique des depots :
)
```

On crée 2 ménages nommés: dupont et legrand.

```
28 //On crée des ménages
29 Menage dupont=new Menage("Dupont");
30 Menage legrand=new Menage("Legrand");
```

10 déchets sont créés au total. On imagine que 5 déchets sont créés par les Dupont et 5 par les Legrand.

```

32      //Les déchets des Dupont
33      Dechet dechet1=new Dechet(TypeDechet.VERRE);
34      Dechet dechet2=new Dechet(TypeDechet.METAL);
35      Dechet dechet3=new Dechet(TypeDechet.METAL);
36      Dechet dechet4=new Dechet(TypeDechet.PLASTIQUE);
37      Dechet dechet5=new Dechet(TypeDechet.VERRE);
38
39
40      //Les déchets des Legrand
41      Dechet dechet6=new Dechet(TypeDechet.METAL);
42      Dechet dechet7=new Dechet(TypeDechet.VERRE);
43      Dechet dechet8=new Dechet(TypeDechet.CARTON);
44      Dechet dechet9=new Dechet(TypeDechet.PLASTIQUE);
45      Dechet dechet10=new Dechet(TypeDechet.METAL);

```

Chaque ménage remplit sa corbeille qui lui est associée. Par exemple, la première instruction représente les Dupont qui remplissent leur sous-corbeille n°2 avec dechet6 (ligne 49).

```

48      //Les ménages déposent des déchets dans leur propre corbeille
49      dupont.remplirCorbeille(dechet6,2); // le deuxième paramètre ne
50      dupont.remplirCorbeille(dechet7,0);
51      dupont.remplirCorbeille(dechet8,3);
52      dupont.remplirCorbeille(dechet9,3);
53      dupont.remplirCorbeille(dechet10,3);
54
55      legrand.remplirCorbeille(dechet1,1);
56      legrand.remplirCorbeille(dechet2,1);
57      legrand.remplirCorbeille(dechet3,3);
58      legrand.remplirCorbeille(dechet4,3);
59      legrand.remplirCorbeille(dechet5,2);

```

On affiche les infos du ménage pour vérifier que les corbeilles sont bien remplies.

```
62 //On affiche les infos des ménages pour vérifier
63 System.out.println(dupont.toString()+"\n");
64 System.out.println(legrand.toString()+"\n \n");
```

```
Menage : (Id : 1, Nom : Dupont, Code d'accès : 71, Point de Fidelité : 0
Liste des bons d'achats :
Corbeille :
  Sous Corbeille n° 0 : {VERRE, Type : 1, Poids : 4},
  Sous Corbeille n° 1 :
  Sous Corbeille n° 2 : {METAL, Type : 3, Poids : 5},
  Sous Corbeille n° 3 : {CARTON, Type : 2, Poids : 3}, {PLASTIQUE, Type : 0, Poids : 2}, {METAL, Type : 3, Poids : 5},
Historique des dépôts : )

Menage : (Id : 2, Nom : Legrand, Code d'accès : 142, Point de Fidelité : 0
Liste des bons d'achats :
Corbeille :
  Sous Corbeille n° 0 :
  Sous Corbeille n° 1 : {VERRE, Type : 1, Poids : 4}, {METAL, Type : 3, Poids : 5},
  Sous Corbeille n° 2 : {VERRE, Type : 1, Poids : 4},
  Sous Corbeille n° 3 : {METAL, Type : 3, Poids : 5}, {PLASTIQUE, Type : 0, Poids : 2},
Historique des dépôts : )
```

Les Dupont créent 2 dépôts qu'ils vont jeter dans la poubelle intelligente qui s'appelle poub4.

Les Legrand créent 3 dépôts qu'ils vont jeter dans diverses poubelles.

Par exemple : le premier dépôt (ligne 67) est constitué des déchets de la sous-corbeille n°0 des Dupont et il a été placé dans la poubelle poub4.

```
66 //Chaque ménage fait plusieurs dépôts
67 dupont.creerDepot(0,poub4); // le prem
68 dupont.creerDepot(3,poub4);
69
70 legrand.creerDepot(2,poub1);
71 legrand.creerDepot(3,poub2);
72 legrand.creerDepot(1,poub3);
```

On affiche le centreA. On perçoit bien les dépôts avec les déchets qui sont rattachés au centreA.

```
75 //On affiche les informations du centre de tri
76 System.out.println("\n"+centreA.toString()+"\n \n");
```

```
Centre de Tri : Paris,
, Type poubelle: 0, Poubelle pleine ? : true,
.240764847, Points gagnés : -1, liste des dechets : {VERRE, Type : 1, Poids : 4} / ,
Type poubelle: 1, Poubelle pleine ? : true,
.313077980, Points gagnés : -2, liste des dechets : {METAL, Type : 3, Poids : 5} , {PLASTIQUE, Type : 0, Poids : 2} / ,
Poubelle: 2, Poubelle pleine ? : true,
.410063665, Points gagnés : -2, liste des dechets : {VERRE, Type : 1, Poids : 4} , {METAL, Type : 3, Poids : 5} / ,
Poubelle: 3, Poubelle pleine ? : true,
.976348231, Points gagnés : -1, liste des dechets : {VERRE, Type : 1, Poids : 4} ]
Points gagnés : 0, liste des dechets : {CARTON, Type : 2, Poids : 3} , {PLASTIQUE, Type : 0, Poids : 2} , {METAL, Type : 3, Poids : 5} / ,
```

On affiche les ménages. On voit que l'historique des dépôts a été mis à jour et que leurs de points de fidélité aussi.

```
78 System.out.println(dupont.toString()+"\n");
79 System.out.println(legrand.toString()+"\n");
80
```

```
Menage : (Id : 1, Nom : Dupont, Code d'accès : 71, Point de Fidelité : -1
Liste des bons d'achats :
Corbeille :
Sous Corbeille n° 0 :
Sous Corbeille n° 1 :
Sous Corbeille n° 2 : {METAL, Type : 3, Poids : 5},
Sous Corbeille n° 3 :
Historique des dépôts : [ Id du Depot : 1, heure du Depot : 2025-03-24T21:03:59.976348231, Points gagnés : -1, Liste des dechets : {VERRE, Type : 1, Poids : 4} ][ Id du Depot : 2, heure du Depot : 2025-03-24T21:04:00.000000000, Points gagnés : -2, Liste des dechets : {METAL, Type : 3, Poids : 5} ][ Id du Depot : 3, heure du Depot : 2025-03-24T21:04:00.000000000, Points gagnés : -1, Liste des dechets : {VERRE, Type : 1, Poids : 4} ]
Menage : (Id : 2, Nom : Legrand, Code d'accès : 142, Point de Fidelité : -5
Liste des bons d'achats :
Corbeille :
Sous Corbeille n° 0 :
Sous Corbeille n° 1 :
Sous Corbeille n° 2 :
Sous Corbeille n° 3 :
Historique des dépôts : [ Id du Depot : 3, heure du Depot : 2025-03-24T21:04:00.240764847, Points gagnés : -1, Liste des dechets : {VERRE, Type : 1, Poids : 4} ][ Id du Depot : 4, heure du Depot : 2025-03-24T21:04:00.000000000, Points gagnés : -2, Liste des dechets : {METAL, Type : 3, Poids : 5} ][ Id du Depot : 5, heure du Depot : 2025-03-24T21:04:00.000000000, Points gagnés : -1, Liste des dechets : {VERRE, Type : 1, Poids : 4} ]
```

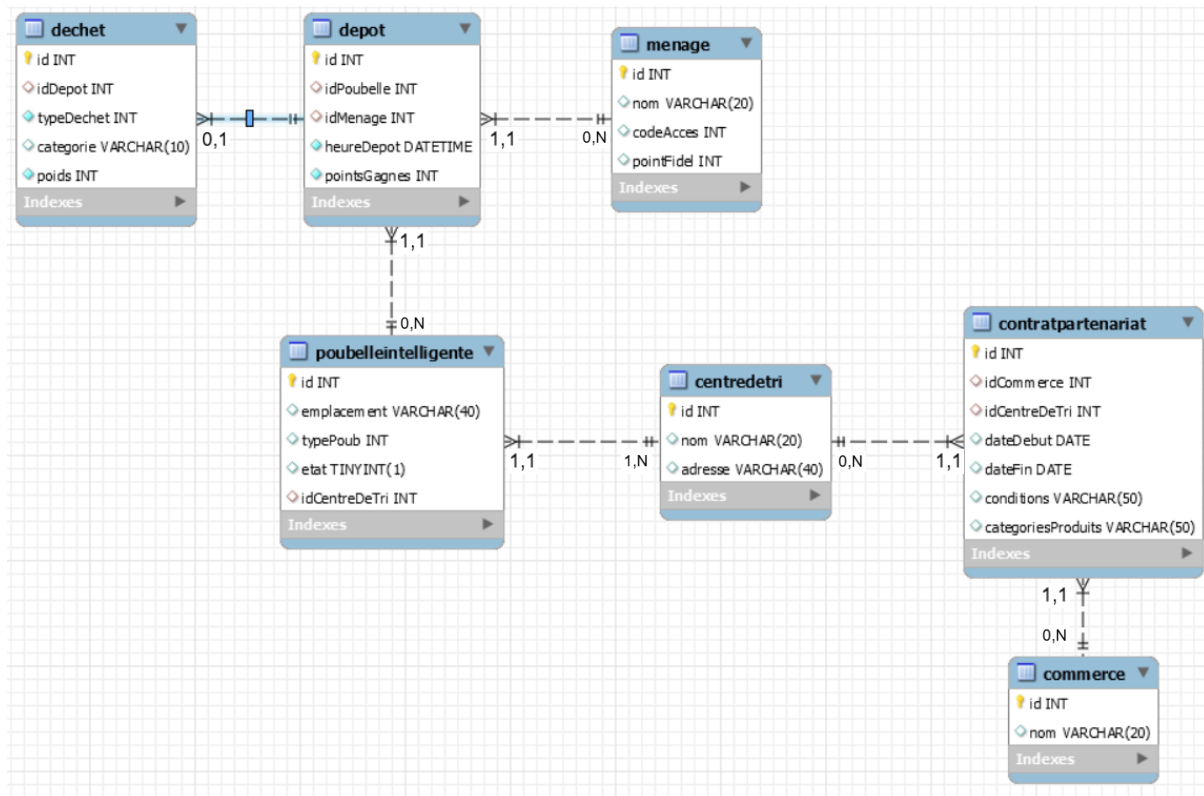
Le centreA décidé de faire la collecte de la poubelle intelligente de id=1. On affiche le centreA et on voit bien que dans la liste de dépôts de celle-ci est vide.

```
81 //On réalise une collecte d'une poubelle dans notre centre de tri
82 centreA.collecter(1);
83 System.out.println("\n"+centreA.toString()+"\n \n");
```

```
( Id du Centre de Tri : 1, Nom du Centre de Tri : CentreA, Adresse du centre de Tri : Paris,
Liste des poubelles :
/ Id de la poubelle : 1, Emplacement poubelle : 3 avenue des Champs-Élysées, Type poubelle: 0, Poubelle pleine ? : false,
Liste des dépôts : VIDE./ ,
```

Création de la base de donnée :

Pour la création de la base de données, nous avons d'abord imaginé le contenu qu'allaient avoir nos tables (en faisant un rapide MCD puis en le convertissant en MLD). Puis nous avons écrit les requêtes SQL qui nous ont permis de créer le tout.



En ce qui concerne la liaison de notre code avec la base de donnée, nous avons suivi des tutoriels en ligne afin d'avoir les drivers nécessaires et de savoir comment adapter des codes dans le cadre de notre projet. Ainsi, nous avons pu définir des méthodes en Java qui nous permettent d'insérer des données, mettre à jour des données ou en supprimer.

Nous avons alors pu insérer des nouvelles lignes dans nos tables à chaque création d'un nouvel objet Java. Et de manière similaire mettre à jour ou supprimer une ligne dans nos tables lorsqu'il y a modification d'attributs sur un de nos objets (exemple : changement d'emplacement ou suppression d'une poubelle).

La base de donnée sur mysql après l'exécution du main

id	nom	adresse
1	CentreA	Paris

id	nom	codeAcces	pointFidel
1	Dupont	71	-1
2	Legrand	142	-5

id	emplacement	typePoub	etat	idCentreDeTri
1	3 avenue des Champs-Élysées	0	0	1
2	13 rue de general de Gaule	1	1	1
3	10 rue Bonaparte	2	1	1
4	23 rue Paul Vaillant Couturier	3	1	1

id	idPoubelle	idMenage	heureDepot	pointsGagnes
1	4	1	2025-03-24 21:31:52	-1
2	4	1	2025-03-24 21:31:52	0
3	1	2	2025-03-24 21:31:52	-1
4	2	2	2025-03-24 21:31:53	-2
5	3	2	2025-03-24 21:31:53	-2

id	idDepot	typeDechet	categorie	poids
1	5	1	VERRE	4
2	5	3	METAL	5
3	4	3	METAL	5
4	4	0	PLASTIQUE	2
5	3	1	VERRE	4
6	NULL	3	METAL	5
7	1	1	VERRE	4
8	2	2	CARTON	3
9	2	0	PLASTIQUE	2
10	2	3	METAL	5

Nos ressentis

Ce deuxième rendu nous a confronté à plusieurs défis, notamment en termes de gestion du temps. Le développement des différentes classes et la mise en place des tests ont pris plus de temps que prévu, ce qui nous a parfois mis sous pression pour respecter les délais. Une autre difficulté majeure a été la compréhension du sujet, en particulier le fonctionnement exact de la collecte des poubelles : nous avons eu du mal à comprendre comment relier les poubelles intelligentes et les différents centres de tri.

Pour les prochains rendus, nous pourrions améliorer notre organisation, notamment en définissant plus précisément notre répartition des tâches et en anticipant davantage les aspects techniques qui pourraient poser problème. Une meilleure clarification du cahier des charges dès le début nous éviterait également des blocages et des pertes de temps inutiles.