**Explication visibilité de chaque attributs et méthodes**

Pour cette modélisation UML, nous avons choisi de faire un diagramme composé de 7 classes UML :

* CentreDeTri
* Poubelle
* Utilisateur
* Corbeille
* Dechet
* Commerce
* ContratPartenariat

**Visibilité des attributs et méthodes pour chaque classe UML**

**1. CentreDeTri**

Cette classe comporte comme attributs (privés, -) :

* + - nom : String
  + - adresse : String

Et comme méthode (publique, +) :

* + + placerPoubelle()
  + + collecterDechets()
  + + genererStatistiques()

Les attributs sont privés (-) pour garantir l'encapsulation et éviter toute modification directe en dehors de la classe.  
Les méthodes sont publiques (+) car elles représentent les actions du centre de tri accessibles par d'autres classes.

**2. Poubelle**

Cette classe comporte comme attributs (privés, -) :

* + – id : int
  + – capaciteMax : int
  + – emplacement : String

Et comme méthode (publique, +) :

* + + identifierUtilisateur()
  + + calculerQuantiteDechets()
  + + verifierCategorieDechets()
  + + attribuerPoints()
  + + notifierCentreTri()

Les attributs sont privés (-) pour empêcher leur modification directe (par exemple : empêcher de changer l’ID d’une poubelle).  
Les méthodes sont publiques (+) car elles doivent être appelées par d'autres classes, comme Utilisateur ou CentreDeTri.

**3. Utilisateur**

Cette classe comporte comme attributs (privés, -) :

* + – id : int
  + – nom : String
  + – pointsFidelite : int

Et comme méthode (publique, +) :

* + + deposerDechets()
  + + consulterHistorique()
  + + convertirPoints()

Les attributs sont privés (-) pour protéger les informations personnelles et éviter la modification directe du nombre de points de fidélité.  
Les méthodes sont publiques (+) car elles permettent à l’utilisateur d’interagir avec le système.

**4. Corbeille**

Cette classe comporte comme attributs (privés, -) :

* + – listeDechets : List<Dechet>

Et comme méthode (publique, +) :

* + + ajouterDechets()
  + + viderCorbeille()

L’attribut listeDechets est privé (-) pour éviter que d'autres classes ne modifient directement son contenu.  
Les méthodes sont publiques (+) car elles permettent d’ajouter ou de vider la corbeille.

**5. Dechet**

Cette classe comporte comme attributs (privés, -) :

* + – type : String
  + – poids : float

Et comme méthode (publique, +) :

* + + getType()
  + + getPoids()

Les attributs sont privés (-) pour empêcher la modification directe du type ou du poids d’un déchet.  
Les méthodes de lecture getType() et getPoids() sont publiques (+) pour que d’autres classes puissent lire ces informations sans pouvoir les modifier directement.

**6. Commerce**

Cette classe comporte comme attributs (privés, -) :

* + – nom : String
  + – categoriesProduit : List<String>

Et comme méthode (publique, +) :

* + + echnagerPoints()

Les attributs sont privés (-) pour éviter des modifications non contrôlées (exemple : éviter que n’importe quelle classe ne modifie les catégories de produits).  
La méthode est publique (+) pour permettre aux utilisateurs d’échanger leurs points contre des réductions.

**7. ContratPartenariat**

Cette classe comporte comme attributs (privés, -) :

* + – dateDebut : Date
  + – dateFin : Date

Et comme méthode (publique, +) :

* + + definirReglesUtilisation()

Les attributs sont privés (-) pour protéger les informations du contrat.  
La méthode est publique (+) pour que le commerce puisse définir les règles d’utilisation.

**Explication cardinalité et choix d’association**

1. **CentreDeTri → gère plusieurs Poubelles**

Le type d’association est une aggregation (\◇). Sa cardinalité est 1 (CentreDeTri) → 0..\* (Poubelle). En effet, un centre de tri possède plusieurs poubelles mais celles-ci peuvent exister indépendamment du centre de tri.

1. **Utilisateur → dépose dans une Poubelle**

Il s’agit ici d’une association simple (—). Sa cardinalité est de 1 (Utilisateur) → 0..\* (Poubelle). En effet, un utilisateur peut déposer ses déchets dans plusieurs poubelles et une poubelle est utilisée par plusieurs utilisateurs.

1. **Poubelle → stocke plusieurs Déchets**

Le type d’association est une composition (◆). Sa cardinalité est de 1 (Poubelle) → 0..\* (Dechet). En effet, une poubelle contient des déchets et ceux-ci n’existent pas sans une poubelle spécifique (si la poubelle disparait, les déchets ne sont plus stockés).

1. **Utilisateur → possède une Corbeille**

Le type d’association est une aggregation (\◇). Sa cardinalité est de 1 (Utilisateur) → 1 (Corbeille). En effet, un utilisateur a une seule corbeille et celle-ci lui appartient mais la corbeille peut exister sans utilisateur.

1. **Corbeille → contient plusieurs Déchets**

Le type d’association est une composition (◆). Sa cardinalité est de 1 (Corbeille) → 0..\* (Dechet). En effet, une corbeille contient plusieurs déchets et ces déchets disparaissent avec la corbeille.

1. **Utilisateur → échange des points avec Commerce**

Il s’agit ici d’une association simple (—). Sa cardinalité est de 1 (Utilisateur) → 0..\* (Commerce). En effet, un utilisateur peut échanger ses points avec plusieurs commerces et un commerce peut avoir plusieurs utilisateurs.

1. **Commerce → définit un ContratPartenariat avec CentreDeTri**

Il s’agit ici d’une association simple (—). Sa cardinalité est de 1 (Commerce) → 1 (ContratPartenariat) → 1 (CentreDeTri). En effet, chaque commerce définit un contrat avec un centre de tri, un contrat lie un seul commerce et un seul centre de tri.