****

**PROJET JAVA**

MARLIER Laura

GRIFFON Antonin

PIGNOL Romain

LARCHER Elya

**Sommaire**

Description des classes et des associations du diagramme UML …………………………. 3

Description des méthodes des classes du diagramme UML ………………………………… 4

Description des classes du diagramme UML

Afin de modéliser les informations nécessaires pour représenter le fonctionnement du tri, nous avons défini onze classes, deux associations de classes, quatre relations de généralisation et deux énumérations.

Les poubelles intelligentes sont définies par un identifiant, un emplacement (une latitude et une longitude), un code d’accès, un volume maximal et un volume total actuel. Une relation d’héritage est créée avec cette classe « Poubelles intelligentes ». Il en découle quatre nouvelles classes : « Poubelle jaune », « Poubelle verte », « Poubelle bleue », « Poubelle classique ».

En effet, il y a quatre types de poubelles intelligentes qui ont toutes l’attribut *type,* qui est statique, pour définir leur couleur (verte pour le verre, bleue pour le papier, jaune pour le métal, le plastique et le carton et noire pour les déchets de la poubelle classique). De plus, chacune de ces classes a un attribut qui contient la quantité du type de déchets présent. La classe « Poubelle verte » a l’attribut contenant la quantité de verre, la classe « Poubelle jaune » a trois attributs quantité pour le carton, le plastique et le métal, la classe « Poubelle bleue » a l’attribut contenant la quantité de papier et la classe « Poubelle classique » a l’attribut contenant la quantité des déchets définis comme autre.

Ces poubelles appartiennent à un centre de tri qui vient les ramasser d’où la relation d’agrégation entre la classe « Poubelles intelligentes » et la classe « Centre de tri ». Un centre de tri est caractérisé par cinq attributs : son nom, son adresse décomposée en numéro de rue, nom de rue, code postal et ville. De plus, un centre de tri collabore avec des commerces, d’où la création de l’association de classe entre la classe « Commerce » et la classe « Centre de tri ». Cette collaboration va être illustrée à travers un contrat défini par une date de début, une date de fin et le type de produit qu’il concerne.

Un commerce est lui décrit par son identifiant, son nom et son adresse qui comme précédemment est décomposée en quatre attributs : numéro de rue, nom de rue, code postal et ville. Il collabore avec les centres de tri afin de proposer des bons de réduction ou des bons d’achat aux ménages qui réalisent un bon tri de leurs déchets.

Un bon de réduction est lui défini par le type de produit auquel il s’applique et par le pourcentage de réduction qu’il donne. Il est généré sur les comptes des ménages d’où l’association avec la classe « Compte ».

Un compte est caractérisé par le nombre de points de fidélité du ménage concerné et du code d’accès donné au ménage pour utiliser les poubelles. La classe est reliée par une association « avoir » avec la classe « Ménage ».

Un ménage est décrit par son nom, son numéro de compte et son adresse découpée en quatre attributs comme pour les centres de tri et les commerces. Il est relié aux poubelles intelligentes par une association de classe « Dépôt ».

Un dépôt est défini par l’identifiant de la poubelle utilisée, l’heure de dépôt, le type et la quantité de déchets déposés ainsi que le nombre de points obtenus suite à ce dépôt, qu’ils soient positifs ou négatifs.

Enfin, nous avons deux énumérations spécifiant les couleurs des poubelles ainsi que les types de déchets possibles.

Description des méthodes des classes du diagramme UML

Afin d’éclaircir notre diagramme, nous allons décrire les différentes actions des méthodes que nous avons créées dans le diagramme pour chaque classe.

* Les méthodes de la classe « Poubelles Intelligentes » :
  + envoiNotif() : permet de notifier le centre de tri sur l’état de remplissage des poubelles,
  + attribuerPoint() : permet de calculer le nombre de point à attribuer pour le dépôt en fonction des déchets triés,
  + identifierUtilisateur() : permet d’identifier le ménage qui fait son tri,
  + estRemplie() : envoie un booléen pour savoir si la poubelle est pleine.

Les autres méthodes sont les accesseurs (getters) et les mutateurs (setters) des attributs *CodeAccès* et *VolumeActuel.*

* Les méthodes de la classe « Centre de tri » :
  + retirerPoub() : permet de retirer une poubelle de tri de son emplacement,
  + ajouterPoub() : permet d’ajouter une poubelle de tri,
  + collecterPoub() : permet de vider une poubelle remplie,
  + Statistics() : permet d’effectuer des statistiques sur les données collectées par les poubelles. Cette méthode pourra être développée dans la suite du projet. Par exemple, elle pourra permettre d’évaluer l’utilité de l’emplacement de certaines poubelles ou encore déterminer si la vitesse de remplissage des poubelles est régulière.
* Les méthodes des classes Poubelle verte, jaune, bleue et classique :
  + DéchetValide() : renvoie un booléen permettant de vérifier si les déchets sont mis dans la bonne poubelle pour par la suite déterminer les points positifs ou négatifs attribués.

Les autres méthodes sont les accesseurs (getters) et les mutateurs (setters) des attributs contenant les quantités de déchets.

* Les méthodes des classes « BonRéduction » sont l’accesseur et le mutateur de l’attribut *pourcentageRéduction.*
* Les méthodes de la classe « Compte » :
  + convertir() : permet de convertir les points de fidélité en bon de réduction, un objet de la classe « BonRéduction »,
  + consulterHist() : permet de consulter l’historique des dépôts.

Les autres méthodes sont les accesseurs (getters) et les mutateurs (setters) des attributs *pointFidélité* et *codeDonné*.

* Les méthodes de la classe « Compte » sont les accesseurs (getters) et les mutateurs (setters) des attributs de la classe.
* Les méthodes de l’association de classe « Dépôt » sont les accesseurs (getters) et les mutateurs (setters) des attributs de l’association de classe.
* Les méthodes de l’association de classe « Contrat » :
  + renouveler() : permet de renouveler le contrat avec des nouvelles dates de début et de fin.

Les autres méthodes sont les accesseurs (getters) et les mutateurs (setters) des attributs de l’association de classe.