

GALBEZ Flavien

GHAHARIAN Alexandre

TD J

Projet de POO : Gestion des dons pour une association

Détails sur le code :

- Toutes les saisies sont sécurisées
- Dans le fichier « Beneficiares.txt », il y a un « : » à la place d'un « ; », on l'a donc corrigé

Le diagramme UML :

Pour commencer nous avons la class `Personne`, qui est mis en abstract car nous ne créerons pas de personne lors de l'exécution du code. Trois class hérite de `Personne`, la class `Adherent`, `Beneficiaire` et `Personne_morale`. Un adhérent peut effectuer un don donc la class `Adherent` et `Don` sont associés, afin que l'objet de l'adhérent puisse être stocké. Avant de pouvoir effectuer un don, il faut d'abord créer un objet et le référencer : savoir quel est le type de l'objet, son montant s'il est vendu et d'autres description si besoin. C'est pourquoi une class `Objet` est associé à la class `Don` pour créer cet objet et le stocker. La class `Objet_volumineux` hérite de la class `Objet` afin de préciser les dimensions de l'objet en question, s'il est jugé trop grand. Lorsque l'objet est créé et stocké, il peut être donné, il y a trois class qui héritent de la class `Don`, `Don_Depot_vente`, `Don_asso` et `Don_garde_meuble`, ces trois class permettent de savoir où l'objet donné doit aller lors de la saisie du don. `Don_Depot_vente` permet de savoir quelle est le montant de l'objet et la date à laquelle l'objet sera vendu et à quel dépôt vente l'objet se trouve. `Don_asso` permet de stocker l'objet directement à l'association.

`Don_garde_meuble` permet de savoir la date d'enlèvement de l'objet, à quel arde meuble l'objet est stocké et le bénéficiaire qui obtiendra l'objet. Comme n'importe quel don est forcément effectué à un bénéficiaire les class `Beneficiaire` et `Don` sont associés. Nous avons aussi lié l'interface `Comparable` à la class `Don`, afin de trier les dons suivant différents critères comme demandé dans l'énoncé.

Pour chaque action souhaitée, il y a un menu associé. D'abord, il y a le menu de démarrage où vous pouvez sélectionner l'action que vous voulez faire. Toutes saisies effectuées dans n'importe quel menu ou questions sont sécurisées pour éviter que l'utilisateur rentre des réponses erronées. À la fin de chaque action, l'utilisateur revient sur le menu qu'il a choisi et il peut sortir du menu, pour retourner dans le menu de démarrage pour sélectionner une autre action ou tout simplement fermer le programme.

La première action du menu de démarrage est le module Personne. Nous pouvons alors sélectionner :

- En 1, la constitution de la liste des adhérents qui sont stockés dans une liste grâce à la fonction public static List<Adherent> Lecture_adherents(string nom_fichier), qui lit le fichier .txt et permet de séparer les informations et les stocker.
- En 2, même principe qu'en 1 mais pour le fichier Beneficiaire.txt avec la fonction public static List<Beneficiaire> Lecture_beneficiares(string nom_fichier).
- En 3, il est possible de chercher un bénéficiaire avec son numéro de téléphone en parcourant le fichier avec cette fonction public static void Recherche_tel(List<Beneficiaire> beneficiaires).
- En 4, même principe qu'en 3 mais avec le nom du bénéficiaire public static void Recherche_nom(List<Beneficiaire> beneficiaires).

La deuxième action du menu de démarrage est le module Don. Nous pouvons alors sélectionner :

- En 1, la création d'un objet pour le référencer et le stocker dans une liste avec la fonction public static Objet Creation_objet(int reference_objet).
- En 2, lorsque les objets sont créés, il est possible de concevoir les dons avec les objets stockés, avec la fonction public static Don Creation_don(List<Adherent> adherents, int id_don), ce qui permet de savoir quel adhérent effectue le don et ce qu'il donne.
- En 3, suivant la liste des dons effectués, grâce à la fonction public static void Stockage_Don(Don don, List<Personne_morale> personnes_morales, List<Don_depot_vente> dons_depot_vente, List<Don_garde_meuble> dons_garde_meuble, List<Don_asso> dons_asso, ArrayList archives, List<Beneficiaire> beneficiaires, List<Adherent> adherents, List<Objet> objets), qui permet de vérifier le don et de l'accepter ou de le refuser, suivant son acceptation, le don sera stocké dans une liste différentes.
- En 4, lorsque le don est accepté, il doit être transférer à un bénéficiaire, cette fonction public static void Transférer_Don(List<Don_depot_vente> dons_depot_vente, List<Don_garde_meuble> dons_garde_meuble, List<Beneficiaire> beneficiaires, List<Don_asso> dons_asso, ArrayList archives, List<Objet> objets) permet de le choisir.
- En 5, le don est effectué et l'objet n'est plus en notre possession, il faut donc archiver la fiche du don et la fonction public static void Archive_Don(List<Don_depot_vente> dons_depot_vente, List<Don_garde_meuble> dons_garde_meuble, List<Don_asso> dons_asso, ArrayList archives), nous le permet.

La troisième action du menu de démarrage est le module Tri. Nous pouvons sélectionner :

- En 1, nous pouvons voir la liste des dons refusés avec public static void Tri_don_refuse(ArrayList archives).
- En 2, la liste des dons qui n'ont pas encore été traité par un adhérent avec public static void Tri_traitement_stock(List<Don_depot_vente> dons_depot_vente, List<Don_garde_meuble> dons_garde_meuble, List<Don_asso> dons_asso).
- En 3, la liste des dons donnés ou vendu avec public static void Tri_don_vendu_donnes(List<Don_garde_meuble> liste_gm).

- En 4, la liste des dons stockés par entrepôt et catégorie avec public static void `Tri_don_par_entrepot_categorie(List<Don_depot_vente> dons_depot_vente)`.
- En 5, la liste des dons par entrepôt et par volume avec public static void `Tri_volume(List<Objet> objets)`.
- En 6, la liste des dons par dépôt-vente et par prix avec public static void `Tri_par_depot_vente_prix(List<Don_depot_vente> dons_depot_vente)`.

La quatrième action du menu de démarrage est le module Statistique. Nous pouvons sélectionner :

- En 1, calculer la moyenne du temps de stockage dans les dépôt-vente avec cette fonction public static double `Moyenne_temps_stock_dv(List<Don_depot_vente> dons_depot_vente)` et dans les garde-meubles avec cette fonction public static double `Moyenne_temps_stock_gm(List<Don_garde_meuble> dons_gm)`.
- En 2, calculer la moyenne des prix en dépôt-vente avec cette fonction public static double `Moyenne_prix_dv(List<Don_depot_vente> dons_dv)`.
- En 3, calculer la moyenne d'âge des bénéficiaires avec cette fonction public static double `Moyenne_age_benef(List<Beneficiaire> beneficiaires)`.

La cinquième action du menu de démarrage est le module Autre. Nous pouvons sélectionner :

- En 1, le pourcentage de dons supprimés avec cette fonction public static double `Nb_supprime(ArrayList archives)`.
- En 2, le pourcentage de dons stockés dans un dépôt-vente avec cette fonction public static double `Nb_dv(ArrayList archives)`.
- En 3, le pourcentage de dons stockés dans un garde-meuble avec cette fonction public static double `Nb_gm(ArrayList archives)`.
- En 4, le Pourcentage de dons stockés dans l'asso ou chez l'adhérent avec cette fonction public static double `Nb_asso(ArrayList archives)`.

La sixième action du menu de démarrage permet de quitter et fermer le programme, ce qui met à terme la simulation.