

Programmation orientée objet

Projet finale : Gestionnaire des dons d'une association caritative

A l'occasion de mon projet final de POO, nous devons réaliser un gestionnaire des dons d'une association caritative. Nous avons trois contraintes majeures :

- Le projet doit répondre aux besoins de l'association.
- Le projet doit contenir tous les modules et fonctions demandée dans le sujet
- Le projet doit utiliser au mieux les concepts d'héritages et de délégation ainsi que d'autres concepts de programmations vue ce semestre

A cela nous ajoutons une contrainte souvent sous-estimée : Nous devons faire le projet a 2. En particulier parce que nous sommes deux élève avec des niveaux très différents, nous avons établi nos propres « convention » afin de mieux nous retrouver.

Nous commenceront par développée les fonctionnalités de notre programme, puis nous parlerons des interprétations et libertés prise vis-à-vis du sujet, enfin nous commenterons la structure du code.

A. Les fonctionnalités :

Par souci de concision, cette partie ne traitera pas de la partie commune à tous les élèves. L'intégralité des fonctions ont été faites sans impasse.

1) Le projet contient des données prête pour tester nos fonctionnalités

Tout d'abord, en lançant la première fois le programme, si vous observer par exemple la liste des dons, celle-ci n'est pas vide. Idem pour toutes les autres données de l'association. Une association appelée « Aider donner agir » a été charger depuis le fichier « Sauvegarde.txt ». Ceci vous permettra de tester les fonctionnalités du programme sans perdre de temps.

2) Quitter en enregistrant

L'intégralité des données de l'association est contenue dans l'objet Association.monAssociation.MonAssociation est une instance comme une autre. On peut ainsi facilement adapter notre programme afin qu'il gère plusieurs associations, cette idée a été abandonné par souci de simplicité. L'avantage reste qu'il suffit de sérialiser « monAssociation » pour l'enregistrer. Au départ, cette instance était non « static » et était créée dans le programme principal. Un paramètre Association se baladait alors avec nous, tout le temps au travers des fonctions. Nous avons décidé de créer une variable global, statatic accessible en tout temps pour rendre notre programme plus clair. Les données sont enregistrées dans Sauvegarde.txt en écrasant la sauvegarde précédente.

3) Notre menu

Pour utiliser notre menu, déplacer vous avec les flèches directionnelles de votre clavier et appuyer sur entrer pour effectuer un choix. À chaque fois que vous appuierez sur une touche, le contenu de la console sera effacé puis affiché de nouveau, donnant l'illusion qu'une barre de sélection se déplace sur le menu.

Le menu est généré grâce à la fonction « Menu.LauchMenu » qui prends un paramètre un titre et une série de choix, et renvoie le numéro d'un choix effectuer. Ainsi, le titre est souvent une question. L'utilisateur ne pourra pas sélectionner le titre qui s'affichera pourtant toujours. Renvoyer le choix sous forme d'un entier permet d'utiliser facilement un « switch » par la suite.

Le menu offre la possibilité de retourner au menu précédent lorsque c'est possible, au menu principal sinon. Peut importe le choix a faire, une petite ligne « Quitter » sera toujours là pour vous. Il s'agit d'appel de la fonction du menu précédent. On pourrait alors s'attendre qu'après avoir fini nos actions, le programme nous ramenerai au point que nous voulions quitter afin de finir l'action en cours. Pour éviter que ce genre d'évènement se produise, le point de sortie du programme ne se situe pas dans la fonction « Program.Main » mais dans « Menu.Quitter ». Cette dernière fonction arrête le processus du programme après avoir sérialisé.

4) Quitter sans enregistrer

Vous penser avoir fait une erreur ? Pas de panique ! Dans le Module Autre, vous trouverez le choix « Quittez sans enregistrer ». Cette fonction met fin au programme sans sérialiser.

5) Modifier le nom de l'association

Certaines personne n'aime pas « Aider Donner Agir » En particulier parce que ADA est également l'acronyme d'une entreprise de location de voiture assez chère et souvent mal entretenue. Pour ces personnes, nous proposons de modifié le nom de l'association.

6) Créé une association

Les militants anti-ADA ne sont pas convaincus par la fonctionnalité précédente, ils trouvent cela trop facile et se sentent méprisé. Nous avons donc ajouté une fonctionnalité qui créé une association et efface toutes les trace de l'existence d'ADA. Pour cela, l'instance Association.monAssociation est remplacé par une nouvelle instance. Toutes les coordonnées de l'association doivent être fournis par l'utilisateur. Toutes les anciennes données sont effacées.

7)Un blindage inébranlable

Dans la seconde partie de la classe menu, vous remarquerez sûrement 350 lignes de fonctions commençant toute par « ask ». Il s'agit de fonction qui demande a l'utilisateur des informations sur un objet. Il en existe de différente sorte :

- Les plus simples telle que « askInt », vérifie que l'entrée est bien un entier entre « min » et « max » avant de le renvoyer. Sinon un message d'erreur s'affiche un message d'erreur et redemande l'information. Ces fonctions n'ont pas été faites a moitié. Pour une date, il n'est pas possible de rentrer le 29 février sauf si l'année est bissextile. L'an 0 n'existe pas.
- Celle qui demande un objet déjà existant comme « askDepot » de vente. Tous les objets du type demandé seront converties grâce à une fonction ToString et l'utilisateur pourra faire son choix dans un menu. Comme l'option « quitter » s'ajouter systématiquement a n'importe quel menu, l'utilisateur n'est pas bloqué l'association ne contient aucune instance du type demandé.
- Enfin certain « Ask » propose à l'utilisateur de créer leur objet. C'est le cas par exemple de « askMarchandise ». Tout choix est toujours présenté sous forme de menu, il n'y a pas moyen de sortir de la classification des objets. Même certaine description tel que « Quelle est la forme de la table ? » font l'objet d'un menu (ronde carree rectangle)

8)Gestion du personnel

Aucune impasse n'a été faite, y compris sur les bonus. Ainsi le bonus proposé pour le Module Personnage est présent. Nous pouvons faire une recherche sur d'autres personnes que les bénéficiaires. Nous pouvons également congédier un membre, exclure un bénéficiaire ou modifier leurs informations personnelles

Si les autres bonus sont très faciles, congédier un membre n'est pas une mince affaire. Si ce membre est donateur ou responsable du dossier d'un don, nous risquons d'avoir des données manquante en le supprimant. La solution trouvée est la création d'un membre fantôme unique a l'association. « fantome » est une fonction, au même titre que président, trésorier, et membre. Créé en même temps que l'association, le fantôme remplace les attributs donateur d'un don ou responsable du dossier lors de la suppression du membre.

9)Suppression des doublons de personnes

Voici une innovation que vous n'avez sûrement vue chez aucun de mes camarades ! Nous avons choisi d'appliquer une suppression de doublons sur les personnes physiques plutôt que sur les dons ou les espaces de stockages parce qu'il vous est possible de chargé une liste d'adhérents et de bénéficiaires. Or, ceci sont déjà inclus dans l'association. Afin de pouvoir faire vos tests tranquillement, nous vous proposons de chargé ses listes de nouveaux, car vous pourrez toujours les supprimer grâce a la fonction de suppression des doublons.

10)Les élections du bureau de l'association.

Après de noires époques dû à l'expérimentation de l'anarchie dictatorial, le comité de normalisation d'Alexandre Bonan et Yann Abou Jaoudé à décider qu'il n'y aurait qu'un seul président et un trésorier. Lors d'une élection, les anciens présidents et trésoriers sont déchargés de leurs fonctions et l'utilisateur choisira leurs successeurs.

B : Nous interprétations et liberté prise par rapport au sujet

1) La personne Morale

Bien que le sujet propose de considérer un espace de stockage tel qu'un dépôt de vente comme un objet composant une instance morale, nous avons choisi de considérer la notion de personne morale tel que le propose le droit français « une personne morale est une entité dotée de la personnalité juridique[...]par exemple : entreprises, associations... ». Ceci permet de profiter de l'héritage de personne. En effet, une personne moral, comme une personne physique, est doté de coordonnées, d'un nom, etc. En revanche, une personne Morale se compose bien en partie d'un inventaire. Nous avons hésité à faire profiter chaque personne d'un inventaire. Les objets auraient ainsi pu être stocké dans l'inventaire d'un bénéficiaire et rendre l'archivage des dons plus claire. Cela dit, au vu du nombre d'adhérent qui aurait eu un inventaire vide pour rien, nous avons préféré garder une approche plus classique.

2)Solde d'un dépôt de vente.

Lors de sa création, un dépôt de vente obtient un taux de commission.

Le montant d'un objet est déterminé au moment du dépôt dans le dépôt de vente conformément au sujet. En revanche, la solde n'est créditée d'un pourcentage du montant qu'au moment de la vente. Ainsi, il est possible de déposer un don en dépôt de vente avec son prix, puis de changer d'avis pour le déplacer ailleurs.

3)Le donateur et sa maison

Nous avons différencié le donateur des adhérents afin de rendre de donateurs plus rapide. Si un membre devient donateur, un constructeur est prévu de donateur est prévu afin que toutes les données de l'adhérent soit transféré.

Un donateur est un adhérent ayant une maison. Sa maison lui permet de stocker des objets.

4) Tout ou partie des informations doivent être imprimable

Cette consigne est écrite dans le sujet. Comme il n'y a aucune information sur le format, ou même la clarté des informations a imprimé, nous avons choisi de rendre l'intégralité des informations imprimable via une « Fiche Association » qu'est le fichier « Sauvegarde.txt » .

5) Le tris par Volume

Une fonction du module trie demande de trier les objets d'un espace de stockage par Volume. Nous avons choisi de les trier par ordre décroissant et d'ajouter la vaisselle (n'ayant pas de volume) à la suite triée par nombre de pièces.

C. Certaines élément de programmation atypique

1) La classification de nos objets est entièrement faite de relation d'héritages.

Certaines classes peuvent paraître inutile, mais il fallait bien différencier les tables par exemple qui ont deux attributs supplémentaires. De plus, cette méthode permet d'utiliser le mot clef « is » qui est bien pratique. Malgré tout, a divers endroit, vous trouverez des « a.GetType()==typeof(b) » Ce genre d'irrégularité est dû à l'hétérogénéité de l'expérience en C# dans notre groupe couplé a une bonne répartition des tâches.

2) Presque tous les attributs sont publics

Au départ, tout nos attributs était privé. Cela dit, un manque de coordination a fait que nous nous sommes retrouvé avec plusieurs accesseurs. Le problème a été assez important pour les attributs de la classe associations, si bien que nous avons décidé de tout passé public. Les identifiants et référence ne sont disponibles qu'en lecture, conformément à la seule demande du sujet a ce propos.

3) Les « Find » utilisent « Lambda », Les « Sort » utilisent « delegate »

Nous n'avions pas tout les deux la même aisance de l'utilisation de Lambda et delegate. Afin d'avoir au moins un exemple de chaque, et pour que nous nous comprenions plus facilement, nous avons établi cette petite convention.

4) « ToString » et « CompleteToString »

La plupart des objets implémentent une fonction « ToString », qui permet d'en faire une description succincte, idéal à afficher dans un menu par exemple. Celle-ci fait au maximum 3 lignes.

« CompleteToString » est utilisé par exemple pour renvoyer une fiche de don, et peut faire plus de 20 lignes.

5) Interface

Nous utilisons des interfaces de manière efficace quand nous trions une « List » grâce à « .Sort » par exemple. Le sujet ne demande pas clairement d'utiliser une interface que nous avons créée, nous l'avons tout de même fait grâce à notre interface « IMoyenne » bien que nous savons que ce n'était pas strictement nécessaire d'utiliser une interface pour cela.

6) Les trie et les recherches se font dans le Menu.

Grâce aux FindAll,Sort, et leurs fonctions déléguées, ces fonctions font rarement plus de 8 lignes affichage compris. Bien sur, si une recherche doit être répété, alors une fonction est créée.

Vous trouverez ici le Diagramme UML complet. Une version png sera jointe au projet, droite et de meilleur qualité. Bonnes vacances.

