

Projet TD7-8

Gestion des dons pour une association

Amine Agoussal, Cécile Amsallem

TD A, A3

Dimanche 19 Mai 2019

Choix & Fonctionnement global.

Chaque type d'objet dispose de sa propre classe, héritant à plusieurs niveaux de la classe Matériel (appelée ainsi afin d'éviter la confusion avec la classe Object). Ces classes disposent de différentes informations selon le type d'objet. Un struct Dimensions a également été créé afin de contenir longueur, largeur, hauteur dans un même objet.

Un don contient la référence de l'instance de l'objet, de sorte que les détails de l'objet associé au don soient accessibles aisément. Une fois légué, un objet DonLegue est créé, contenant de la même façon l'instance initiale du don ainsi que du bénéficiaire concerné, avec des informations supplémentaires.

Les personnes sont également définies sur une relation d'héritage.

Afin de maintenir une traçabilité, des personnes et des dons, la classe personne contient une liste statique de toutes les personnes existantes, mise à jour lors des créations/suppressions de personnes. La classe Don contient une liste statique des dons disponibles, une liste des dons archivés (refusés ou légués), et une file d'attente de dons à traiter (accepter ou non) par un adhérent.

De même la classe LieuStockage comporte une liste statique des lieux de stockage existants, et les instances de stockages contiennent elles-mêmes une SortedList de dons stockés et une SortedList de dons légués, triés par identifiants. (Nous l'avions initialement triée avec la date de dépôt comme key, mais dans le contexte de ce devoir, l'utilisateur peut rentrer manuellement une date, elle n'est pas générée automatiquement au moment du dépôt, pour pouvoir tester notamment les fonctions de statistiques, ce qui ne garantit plus l'unicité de la date. En condition réelles la date aurait été un choix plus pertinent que l'identifiant au vu des stats à réaliser.)

En termes de collections, un dictionnaire est aussi utilisé, notamment pour le compte des types d'objets par fréquence.

Les personnes implémentent l'interface IComparable<Personne>, ce qui permet de comparer des personnes, sur leur nom par défaut. La majorité des classes implémentent également une interface que l'on a créé : Identifiable, qui assure la disponibilité d'un identifiant pour les divers éléments accessibles.

Les fonctions de statistiques, tri, etc sont toutes utilisées sous la forme des delegate. Combinées avec une fonction InteractionUtilisateur qui retourne le bon delegate selon le choix utilisateur, cela permet de rendre le code des menus clair et compact.

La plupart des méthodes sont génériques et permettent par exemple de spécifier le type de matériel recherché pour un don, en plus du prédicat de recherche.

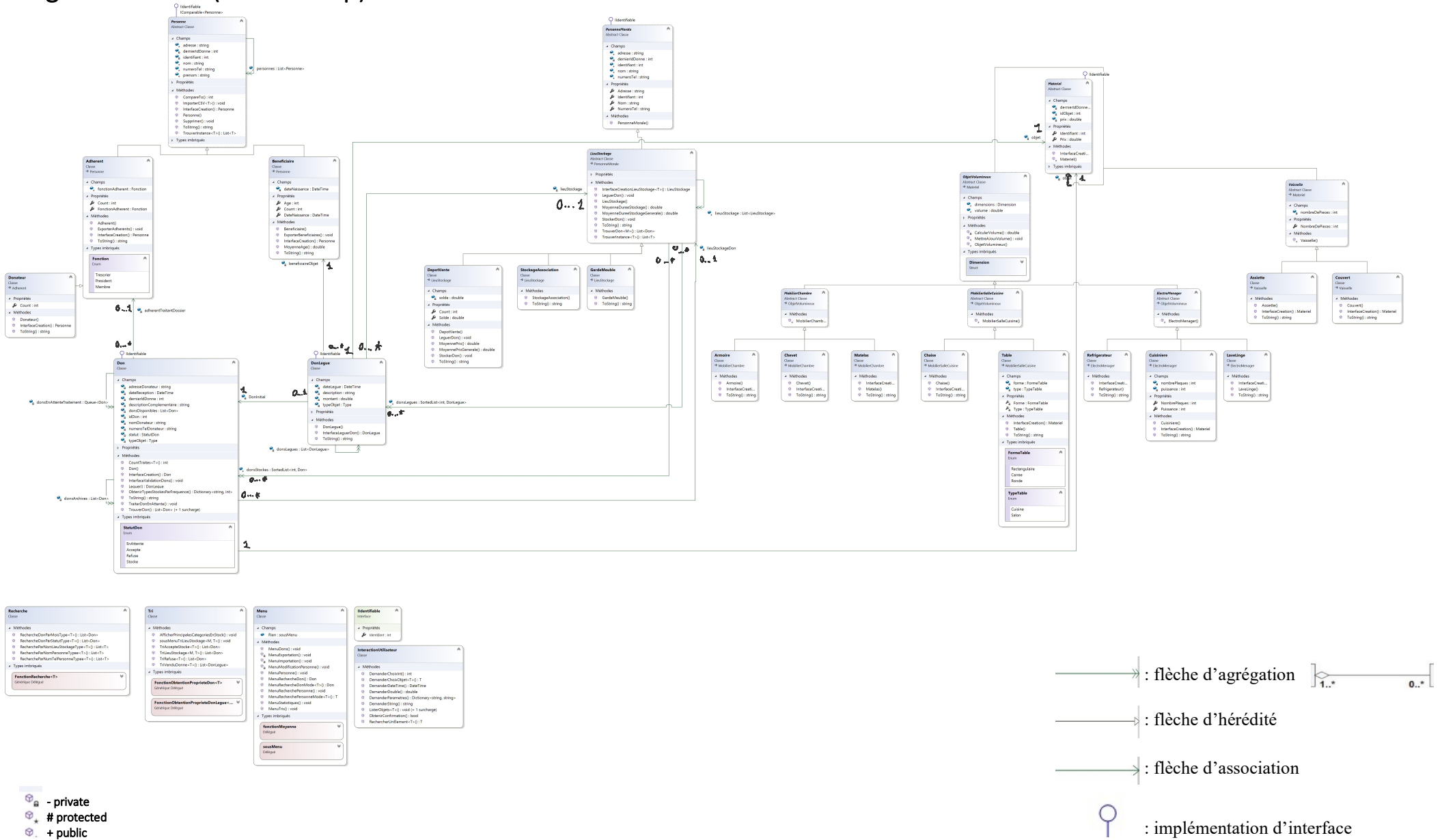
Pour les fonctions du module Autre, nous avons choisi de les répartir dans les menus les plus adéquats. Ainsi vous trouverez dans le menu Personne la possibilité d'exporter les adhérents ou les bénéficiaires sous format .txt. Cette fonction (ExporterBeneficiaire ou ExporterAdherent) exporte les données dans un fichier se trouvant dans le répertoire courant de l'exécutable, afin que chacun puisse facilement retrouver le fichier. De même dans le module Statistique, nous avons ajouté une fonction permettant d'avoir le nombre d'adhérent en plus de celles donnant le nombre de bénéficiaires et de donateurs. Nous avons également donné une autre possibilité dans le module Don : nous pouvons rechercher des dons par type d'objet, et par mois/type en fonction de ce que demande l'utilisateur. Il y a donc bien les quatre fonctionnalités du module Autre, mais elles sont réparties dans les menus correspondants. Les bonus de recherche de personnes et de modification suppression de celles-ci sont également inclus.

A propos du module Statistique, nous avons créé les options demandées dans le module en considérant que la moyenne de temps entre la réception et le retrait des dons se faisait sur toutes les zones de stockage, ce qui nous paraissait plus pertinent. Nous utilisons donc une moyenne générale. Nous avons fait le même choix pour la moyenne portant sur le prix des objets dans les dépôts ventes, nous considérons la moyenne sur tous les dépôts ventes et pas seulement un seul, néanmoins chaque instance de dépôt vente dispose d'une fonction de moyenne de solde. A ce module nous avons ajouté toutes les fonctionnalités du paragraphe statistique demandé. Par ailleurs nous n'avons pas traité la question

« Volume des ventes ventilé suivant les destinations possibles » puisque nous n'en comprenons pas l'intitulé et qu'aucun de nos camarades n'a su nous renseigner dans le peu de temps qu'il nous restait.

Pour ce qui est du module Personne, nous avons donné la possibilité de lire des fichiers autres que ceux fournis avec le sujet du projet. Les fichiers fournis dès l'origine sont automatiquement lus à l'exécution du code. Ainsi l'utilisateur peut ajouter facilement des bénéficiaires ou des adhérents. Nous avons donc dans ce module la fonction issue du module Autre expliquée ci-dessus.

Diagramme UML (Zoommez svp)



Le lecteur PDF semble un peu lent à faire la mise au point lors du zoom, le diagramme est donc disponible au format png également dans le dossier.
(sans légende ou cardinalités toutefois)

Index hiérarchique

Hiérarchie des classes

IComparable	
Personne	19
Adherent.....	6
Donateur.....	12
Beneficiaire	7
IIidentifiable	13
Don.....	9
DonLegue.....	12
Materiel	17
ObjetVolumineux.....	19
ElectroMenager.....	13
Cuisiniere.....	9
LaveLinge	16
Refrigerateur	21
MobilierChambre	18
Armoire.....	7
Chevet.....	8
Matelas.....	17
MobilierSalleCuisine	19
Chaise	8
Table	22
Vaisselle	25
Assiette	7
Couvert	8
Personne	19
PersonneMorale.....	20
LieuStockage.....	16
DepotVente	9
GardeMeuble	13

Documentation des classes

Adherent

Documentation des énumérations membres

enum [Fonction](#) [strong]

Enum contenant les fonctions possibles des adhérents. Accédé avec
`Adherent.Fonction.X`

Valeurs énumérées:

Tresorier	
President	
Membre	

Documentation des constructeurs et destructeur

[Adherent](#) ([Fonction](#) *fonction*, string *nom*, string *prenom*, string *adresse*, string *numeroTel*)

Crée une instance de la classe Adhérent, selon son statut

Cet exemple montre comment initialiser une instance de la classe [Adherent](#)

```
Adherent jacques = new Adherent (Adherent.Fonction.Membre, "DUPONT", "Jacques", "1 Rue  
du filou, Paris", "0123456789");
```

Paramètres:

<i>fonction</i>	Fonction de l'adhérent. Sous la forme Adherent .Fonction.X

Documentation des fonctions membres

void ExporterAdherents ()

static new [Personne](#) InterfaceCreation () [static]

override string ToString ()

Documentation des données membres

[Fonction](#) fonctionAdherent [protected]

Fonction de l'adhérent. Peut évoluer

Documentation des propriétés

`new int Count[static], [get]`

[Fonction](#) `FonctionAdherent[get], [set]`

Armoire

Fonctions membres publiques

- [Armoire](#) (double longueur, double largeur, double hauteur, double [prix](#)=0)
Permet de créer une instance de classe [Armoire](#)
- override string [ToString](#) ()

Fonctions membres publiques statiques

- static new [Materiel InterfaceCreation](#) ()
Permet de créer une armoire à partir des informations rentrées au clavier par l'utilisateur.
-

Assiette

Fonctions membres publiques

- [Assiette](#) (int [nombreDePieces](#), double [prix](#)=0)
Permet de créer une instance de classe [Assiette](#)
- override string [ToString](#) ()

Fonctions membres publiques statiques

- static new [Materiel InterfaceCreation](#) ()
Permet de créer une instance [Assiette](#) à partir des informations rentrées au clavier par l'utilisateur.
-

Beneficiaire

Fonctions membres publiques

- [Beneficiaire](#) (DateTime [dateNaissance](#), string [nom](#), string [prenom](#), string [adresse](#), string [numeroTel](#))
Crée une instance de la classe [Beneficiaire](#)
- override string [ToString](#) ()
- void [ExporterBeneficiaires](#) ()

Fonctions membres publiques statiques

- static double [MoyenneAge](#) ()
- static new [Personne InterfaceCreation](#) ()

Propriétés

- DateTime [DateNaissance](#) [get]

- static new int [Count](#) [get]
- int [Age](#) [get]

Accesseur pour obtenir l'age selon l'année actuelle à l'exécution. Ne calcule qu'avec les années sans se préoccuper du mois / jour

Attributs privés

- DateTime [dateNaissance](#)

Chaise

Fonctions membres publiques

- [Chaise](#) (double longueur, double largeur, double hauteur, double [prix](#)=0)
Permet de créer une instance de classe [Chaise](#)
- override string [ToString](#) ()

Fonctions membres publiques statiques

- static new [Materiel InterfaceCreation](#) ()
Permet d'initialiser la création d'une instance de classe [Chaise](#) à partir des informations rentrées au clavier.

Chevet

Fonctions membres publiques

- [Chevet](#) (double longueur, double largeur, double hauteur, double [prix](#)=0)
Permet de créer une instance de classe [Chevet](#)
- override string [ToString](#) ()

Fonctions membres publiques statiques

- static new [Materiel InterfaceCreation](#) ()
Crée une instance de classe [Chevet](#) à partir des informations rentrées par l'utilisateur.

Couvert

Fonctions membres publiques

- [Couvert](#) (int [nombreDePieces](#), double [prix](#)=0)
Permet de créer une instance de la classe [Couvert](#).
- override string [ToString](#) ()

Fonctions membres publiques statiques

- static new [Materiel InterfaceCreation](#) ()
Initialise la création d'une instance de la classe [Couvert](#) à partir des informations données par l'utilisateur.

Cuisiniere

Fonctions membres publiques

- [Cuisiniere](#) (int [puissance](#), int [nombrePlaques](#), double longueur, double largeur, double hauteur, double [prix](#))
- override string [ToString](#) ()

Fonctions membres publiques statiques

- static new [Materiel InterfaceCreation](#) ()

Propriétés

- int [NombrePlaques](#) [get]
- int [Puissance](#) [get]

Attributs privés

- int [puissance](#)
 - int [nombrePlaques](#)
-

DepotVente

Fonctions membres publiques

- [DepotVente](#) (string [nom](#), string [adresse](#), string [numeroTel](#))
A la création du dépôt vente, son solde est de 1000euros
- double [MoyennePrix](#) ()
Permet de calculer la moyenne du prix des objets stockés dans un dépôt-vente.
- override void [LeguerDon](#) ([DonLegue](#) donLegue)
Permet d'enlever un don des dons stockés et de l'ajouter à la liste des dons légués.
- override void [StockerDon](#) ([Don](#) donAAjouter)
Permet de stocker un don dans le stockage
- override string [ToString](#) ()

Fonctions membres publiques statiques

- static double [MoyennePrixGenerale](#) ()
Permet de calculer la moyenne du prix des objets stockés dans tous les dépôts-ventes.

Propriétés

- double [Solde](#) [get]
- static int [Count](#) [get]

Attributs privés

- double [solde](#)
-

Don

[Don](#) effectué à l'association

Types publics

- enum [StatutDon](#) { [EnAttente](#), [Accepte](#), [Refuse](#), [Stocke](#) }

Fonctions membres publiques

- [Don](#) ([Materiel](#) materielDonne, [Donateur](#) donateur, DateTime [dateReception](#), string [descriptionComplementaire](#)="")
- [DonLegue Leguer](#) ([Beneficiaire](#) beneficiaire, DateTime dateLeguee)
Permet de léguer un don à un bénéficiaire. On crée un [DonLegue](#) avec le [Don](#) et le Bénéficiaire. Supprime également le don de la liste de dons disponibles et l'ajoute aux archives. Supprime également le don du lieu de stockage s'il en avait un.
- override string [ToString](#) ()

Fonctions membres publiques statiques

- static List< [Don](#) > [TrouverDon](#) (Predicate< [Don](#) > predicate)
Renvoie la liste d'instance don ayant la condition passée en argument.
- static List< [Don](#) > [TrouverDon< T >](#) (Predicate< [Don](#) > predicate)
Renvoie la liste de don de type T ayant la condition passée en argument.
- static int [CountTraites< T >](#) (bool InclureArchive=false, [StatutDon](#)[] statutsACompter=null)
Fonction qui compte les dons de type T ayant certains statuts listés en argument. Elle compte également les dons correspondants dans les archives lorsqu'on le précise en argument. Par défaut, elle considère la liste de tous les statuts, donc elle compte tous les dons de type T.
- static Dictionary< string, int > [ObtenirTypesStockesParFrequence](#) ()
- static void [TraiterDonEnAttente](#) ([Adherent adherentTraitantDossier](#), [StatutDon](#) nouveauStatut, [LieuStockage lieuStockageDon](#)=null)
Traite le premier don dans la file d'attente, l'en retire et l'ajoute dans la liste adaptée (dons disponibles ou archivés en cas de refus..
- static void [InterfaceValidationDons](#) ()
Interface de traitement de dons, qui affiche le dernier traitement en attente dans la file, demande les informations à l'utilisateur, puis traite le don en l'enlevant de la file d'attente des dons en attente
- static [Don](#) [InterfaceCreation](#) ()
Interface de création d'un don, appelant les interfaces de créations de [Materiel](#), [Donateur](#). Demande à l'utilisateur une date de reception et une description optionnelle

Propriétés

- int [Identifiant](#) [get]
- string [DescriptionComplementaire](#) [get]
- DateTime [DateReception](#) [get]
- Type [TypeObjet](#) [get]
- [Materiel](#) [Objet](#) [get]
- string [NomDonateur](#) [get]
- string [NumeroTelDonateur](#) [get]
- string [AdresseDonateur](#) [get]
- [StatutDon](#) [Statut](#) [get]
- [Adherent](#) [AdherentTraitantDossier](#) [get]
- [LieuStockage](#) [LieuStockageDon](#) [get]

Attributs privés

- int [idDon](#)
- string [descriptionComplementaire](#)
- DateTime [dateReception](#)
- string [nomDonateur](#)
- string [numeroTelDonateur](#)

- string [adresseDonateur](#)
- Type [typeObjet](#)
- [Materiel](#) objet
- [StatutDon](#) statut
- [Adherent](#) [adherentTraitantDossier](#)
- [LieuStockage](#) [lieuStockageDon](#)

Attributs privés statiques

- static List< [Don](#) > [donsDisponibles](#) = new List<[Don](#)>()
- static List< [Don](#) > [donsArchives](#) = new List<[Don](#)>()
- static Queue< [Don](#) > [donsEnAttenteTraitement](#) = new Queue<[Don](#)>()
- static int [dernierIdDonne](#) = 0

Documentation des fonctions membres

static int CountTraites< T > (bool *InclureArchive* = false, [StatutDon](#) [] *statutsACompter* = null) [static]

Fonction qui compte les dons de type T ayant certains statuts listés en argument. Elle compte également les dons correspondants dans les archives lorsqu'on le précise en argument. Par défaut, elle considère la liste de tous les statuts, donc elle compte tous les dons de type T.

Contraintes de type

T : Materiel

static [Don](#) InterfaceCreation () [static]

Interface de création d'un don, appelant les interfaces de créations de [Materiel](#), [Donateur](#). Demande à l'utilisateur une date de reception et une description optionnelle

static void InterfaceValidationDons () [static]

Interface de traitement de dons, qui affiche le dernier traitement en attente dans la file, demande les informations à l'utilisateur, puis traite le don en l'enlevant de la file d'attente des dons en attente

[DonLegue](#) Leguer ([Beneficiaire](#) *beneficiaire*, DateTime *dateLeguee*)

Permet de léguer un don à un bénéficiaire. On crée un [DonLegue](#) avec le [Don](#) et le Bénéficiaire. Supprime également le don de la liste de dons disponibles et l'ajoute aux archives. Supprime également le don du lieu de stockage s'il en avait un.

static Dictionary<string, int> ObtenirTypesStockesParFrequence () [static]

static void TraiterDonEnAttente ([Adherent](#) *adherentTraitantDossier*, [StatutDon](#) *nouveauStatut*, [LieuStockage](#) *lieuStockageDon* = null) [static]

Traite le premier don dans la file d'attente, l'en retire et l'ajoute dans la liste adaptée (dons disponibles, ou archivés en cas de refus)

static List<[Don](#)> TrouverDon (Predicate< [Don](#) > *predicate*) [static]

Renvoie la liste d'instance don ayant la condition passée en argument.

static List<[Don](#)> [TrouverDon](#)< T > (Predicate< [Don](#) > *predicate*) [static]

Renvoie la liste de don de type T ayant la condition passée en argument.

Contraintes de type

T : Matériel

Donateur

Fonctions membres publiques

- [Donateur](#) ([Fonction](#) fonction, string [nom](#), string [prenom](#), string [adresse](#), string [numeroTel](#))
Crée une instance de la classe [Donateur](#), selon son statut
- override string [ToString](#) ()

Fonctions membres publiques statiques

- static new [Personne](#) [InterfaceCreation](#) ()
Permet de créer un donateur à partir des informations rentrées au clavier par l'utilisateur.

Propriétés

- static new int [Count](#) [get]
-

DonLegue

Classe représentant un don qui a été légué par l'association (donné ou vendu) à un bénéficiaire.

Fonctions membres publiques

- [DonLegue](#) ([Don](#) DonALeguer, [Beneficiaire](#) [beneficiaireObjet](#), DateTime [dateLegue](#))
Constructeur de donLegue. Retirer l'élément de la liste des dons dispos doit être fait après l'avoir légué - ie pour leguer un don appeler don.Leguer !!
- override string [ToString](#) ()

Fonctions membres publiques statiques

- static [DonLegue](#) [InterfaceLeguerDon](#) ()
Interface utilisateur pour leguer un don disponible

Propriétés

- static List< [DonLegue](#) > [DonsLegues](#) [get]
- [LieuStockage](#) [LieuStockage](#) [get]
- string [Description](#) [get]
- double [Montant](#) [get]
- Type [Type](#) [get]
- [Materiel](#) [Objet](#) [get]
- [Beneficiaire](#) [BeneficiaireObjet](#) [get]
- [Don](#) [RefDonInitial](#) [get]
- DateTime [DateLegue](#) [get]
- int [Identifiant](#) [get]

Attributs privés

- [Materiel](#) objet
- [Type](#) typeObjet
- [Don](#) DonInitial
- string [description](#)
- [Beneficiaire](#) beneficiaireObjet
- [LieuStockage](#) lieuStockage
- double [montant](#)
- DateTime [dateLegue](#)

Attributs privés statiques

- static List< [DonLegue](#) > [donsLegues](#) = new List<[DonLegue](#)>()
-

ElectroMenager

Fonctions membres protégées

- [ElectroMenager](#) (double longueur, double largeur, double hauteur, double [prix](#)=0)
-

GardeMeuble

Fonctions membres publiques

- [GardeMeuble](#) (string [nom](#), string [adresse](#), string [numeroTel](#))
 - override string [ToString](#) ()
-

Identifiable

Propriétés

- int [Identifiant](#) [get]
-

InteractionUtilisateur

Fonctions membres publiques statiques

- static bool [ObtenirConfirmation](#) (string message="Vous confirmez ?")
Demande une confirmation à l'utilisateur
- static void [ListerObjets< T >](#) (string message, T[] listeChoix)
Affiche une array
- static void [ListerObjets< T >](#) (string message, List< T > listeChoix)
- static int [DemanderChoixInt](#) (string message, string[] listeIntitulesChoix)
Affiche un message à l'utilisateur, et lui propose des choix parmi lesquels il doit en rentrer un.
- static string [DemanderString](#) (string message)
Affiche un message à l'utilisateur, et lui demande une entrée
- static T [DemanderChoixObjet< T >](#) (string message, T[] listeChoix, string[] listeIntitulesChoix=null)
Affiche un message à l'utilisateur, et lui propose des choix parmi lesquels il doit en choisir un.
- static Dictionary< string, string > [DemanderParametres](#) (string[] parametresADemander)

Demande un par un plusieurs paramètres à l'utilisateur

- static DateTime [DemanderDateTime](#) (string message="Entrez une date")

Demande une date à l'utilisateur

- static double [DemanderDouble](#) (string message="Entrez une nombre")

Demande un double à l'utilisateur

- static T [RechercherUnElement< T >](#) (Recherche.FonctionRecherche< T > fonctionDeRecherche, bool demanderChoix=true, string nomArgumentRecherche=null)

Demande à l'utilisateur une entrée pour une recherche, et lui permet de choisir un objet parmi les résultats de cette recherche.

Documentation des fonctions membres

static int DemanderChoixInt (string *message*, string [] *listeIntitulesChoix*) [static]

Affiche un message à l'utilisateur, et lui propose des choix parmi lesquels il doit en rentrer un.

Paramètres:

<i>message</i>	Message à afficher pour expliquer le contexte à l'utilisateur
<i>listeIntitulesChoix</i>	Liste d'intitulés de choix possibles

Renvoie:

entier entre 0 et le nombre de choix possibles -1, représentant l'index du choix dans la liste de strings.

static T DemanderChoixObjet< T > (string *message*, T [] *listeChoix*, string [] *listeIntitulesChoix* = null) [static]

Affiche un message à l'utilisateur, et lui propose des choix parmi lesquels il doit en choisir un.

Template Parameters:

<i>T</i>	Type des objets choisis à retourner
----------	-------------------------------------

Paramètres:

<i>message</i>	Message à afficher pour expliquer le contexte à l'utilisateur
<i>listeChoix</i>	Objets parmi lesquels l'utilisateur peut choisir
<i>listeIntitulesChoix</i>	Optionnel : Une liste d'intitulé de choix. Si elle n'est pas renseignée, utilise les représentations textuelles des objets comme intitulés.

Renvoie:

L'objet choisi

static DateTime DemanderDateTime (string *message* = "Entrez une date") [static]

Demande une date à l'utilisateur

Paramètres:

<i>message</i>	Message de contexte à afficher
----------------	--------------------------------

Renvoie:

Element DateTime correspondant à l'entrée utilisateur

```
static double DemanderDouble (string  message =  
"Entrez une nombre")[static]
```

Demande un double à l'utilisateur

Paramètres:

<i>message</i>	Message de contexte à afficher
----------------	--------------------------------

Renvoie:

double correspondant à l'entrée utilisateur

```
static Dictionary<string, string> DemanderParametres (string []  
parametresADemander)[static]
```

Demande un par un plusieurs paramètres à l'utilisateur

Paramètres:

<i>parametresADemander</i>	array des noms des parametres à demander. Seront utilisés comme clés du dico et affichés à l'utilisateur lors de la demande
----------------------------	---

Renvoie:

Dico de clés paramètres et valeurs les valeurs entrées par l'utilisateur

```
static string DemanderString (string  message)[static]
```

Affiche un message à l'utilisateur, et lui demande une entrée

Paramètres:

<i>message</i>	Message à afficher pour expliquer le contexte à l'utilisateur
----------------	---

Renvoie:

string rentré par l'utilisateur.

```
static void ListerObjets< T > (string  message, T []  listeChoix)[static]
```

Affiche une array

Paramètres:

<i>message</i>	Message à afficher à l'utilisateur
<i>listeChoix</i>	array à afficher

```
static void ListerObjets< T > (string  message, List< T >  listeChoix)[static]
```

```
static bool ObtenirConfirmation (string  message =  
"Vous confirmez ?")[static]
```

Demande une confirmation à l'utilisateur

Paramètres:

<i>message</i>	Message à afficher à l'utilisateur
----------------	------------------------------------

Renvoie:

true ou false selon la confirmation de l'utilisateur

```
static T RechercherUnElement< T > (Recherche.FonctionRecherche< T >  
fonctionDeRecherche, bool demanderChoix = true, string nomArgumentRecherche  
= null) [static]
```

Demande à l'utilisateur une entrée pour une recherche, et lui permet de choisir un objet parmi les résultats de cette recherche.

Template Parameters:

<i>T</i>	Type des objets choisis à retourner
----------	-------------------------------------

Paramètres:

<i>fonctionDeRecherche</i>	Fonction de recherche prenant une entrée en argument .
<i>demanderChoix</i>	booléen indiquant si un choix est requis ou non. Si non, la fonction retourne toujours une valeur nulle/par défaut
<i>nomArgumentRecherche</i>	nom de l'argument de recherche. Doit renseigner l'utilisateur sur l'entrée qui lui est demandée, par exemple, recherche par "titre", par "ville", etc.

Renvoie:

LaveLinge

Fonctions membres publiques

- [LaveLinge](#) (double longueur, double largeur, double hauteur, double [prix](#))
- override string [ToString](#) ()

Fonctions membres publiques statiques

- static new [Materiel](#) [InterfaceCreation](#) ()
-

LieuStockage

Fonctions membres publiques

- [LieuStockage](#) (string [nom](#), string [adresse](#), string [numeroTel](#))
- virtual void [StockerDon](#) ([Don](#) donAAjouter)
Ajoute un don à la liste des dons stockés.
- virtual void [LeguerDon](#) ([DonLegue](#) donLegue)
Permet d'enlever un don des dons stockés et de l'ajouter à la liste des dons légués.
- List< [Don](#) > [TrouverDon](#)< M > (Predicate< [Don](#) > predicate)
Permet de trouver les dons d'un certain type correspondants à la condition passée en argument.
- double [MoyenneDureeStockage](#) ()
Permet de calculer la durée de stockage moyenne d'un lieu de stockage.
- override string [ToString](#) ()

Fonctions membres publiques statiques

- static List< T > [TrouverInstance< T >](#) (Predicate< T > predicat)
Trouve les instances de lieu de stockage correspondant à la condition donnée
- static double [MoyenneDureeStockageGenerale](#) ()
Permet de calculer la durée de stockage moyenne de tous les lieux de stockage.
- static [LieuStockage](#) [InterfaceCreationLieuStockage< T >](#) ()
Permet de créer un lieu de stockage selon les saisies en console faites par l'utilisateur.

Attributs protégés

- SortedList< int, [Don](#) > [donsStockes](#)
- SortedList< int, [DonLegue](#) > [donsLegues](#)

Attributs protégés statiques

- static List< [LieuStockage](#) > [lieuxStockage](#) = new List<[LieuStockage](#)>()

Propriétés

- SortedList< int, [Don](#) > [DonsStockes](#) [get]
- SortedList< int, [DonLegue](#) > [DonsLegues](#) [get]

Matelas

Fonctions membres publiques

- [Matelas](#) (double longueur, double largeur, double hauteur, double [prix](#)=0)
- override string [ToString](#) ()

Fonctions membres publiques statiques

- static new [Materiel](#) [InterfaceCreation](#) ()

Materiel

[Materiel](#) simple que l'association peut recevoir/donner/vendre. Ses propriétés dépendent du type d'objet et sont donc propres aux classes abstraites

Fonctions membres publiques

- delegate [Materiel](#) [CreateurMateriel](#) ()
Delegate ayant pour but la création de matériel.

Fonctions membres publiques statiques

- static [Materiel](#) [InterfaceCreation](#) ()
Interface permettant la création d'un objet en fonction du type voulu par l'utilisateur. Renvoie aux interfaces de créations de chaque objet précis

Fonctions membres protégées

- [Materiel](#) (double [prix](#))
Constructeur "automatique" de la classe. A chaque objet instancié, on lui donne un id correspondant à son "numéro" d'objets, [prix](#) =0

Paramètres:

<i>prix</i>	prix de l'objet
-------------	-----------------

Propriétés

- int [Identifiant](#) [get]
- double [Prix](#) [get]

Attributs privés

- int [idObjet](#)
- double [prix](#)

Attributs privés statiques

- static int [dernierIdDonne](#) = 0

Menu**Fonctions membres publiques**

- delegate void [sousMenu](#) ()
Delegate sous menu qui ne prend ni argument et ne retourne rien. Utile pour afficher qqchse ou ouvrir un menu.
- delegate double [fonctionMoyenne](#) ()
Delegate de fonction de moyenne

Fonctions membres publiques statiques

- static void [MenuRecherchePersonne](#) ()
- static void [MenuTris](#) ()
- static void [MenuPersonne](#) ()
- static [Don MenuRechercheDon](#) (bool demanderChoix)
- static [Don MenuRechercheDonMode< T >](#) (bool demanderChoix)
- static void [MenuStatistiques](#) ()
- static T [MenuRecherchePersonneMode< T >](#) (bool demanderChoix)
- static void [MenuDons](#) ()

Attributs publics statiques

- static [sousMenu Rien](#) = () => { }
Sous menu "vide" permettant ainsi de retourner à l'accueil sans rien faire.

Fonctions membres privées statiques

- static void [MenuModificationPersonne](#) ()
- static void [MenuImportation](#) ()

MobilierChambre**Fonctions membres protégées**

- [MobilierChambre](#) (double longueur, double largeur, double hauteur, double [prix](#)=0)

MobilierSalleCuisine

Fonctions membres protégées

- [MobilierSalleCuisine](#) (double longueur, double largeur, double hauteur, double [prix](#)=0)
-

ObjetVolumineux.Dimension

Attributs publics

- double [longueur](#)
 - double [largeur](#)
 - double [hauteur](#)
-

ObjetVolumineux

Classes

- struct [Dimension](#)

Fonctions membres protégées

- [ObjetVolumineux](#) (double longueur, double largeur, double hauteur, double [prix](#))
Constructeur d'un objet volumineux depuis ses dimensions données
- void [MettreAJourVolume](#) ()
Méthode d'instance qui met à jour l'attribut du volume de l'objet depuis ses dimensions

Attributs protégés

- [Dimension](#) [dimensions](#)
- double [volume](#)

Propriétés

- [Dimension](#) [Dimensions](#) [get]
- double [Volume](#) [get]

Fonctions membres privées statiques

- static double [CalculerVolume](#) ([Dimension](#) [dimensions](#))
Méthode de classe calculant un volume depuis des dimensions
Méthode d'instance qui met à jour l'attribut du volume de l'objet depuis ses dimensions
-

Personne

Fonctions membres publiques

- [Personne](#) (string [nom](#), string [prenom](#), string [adresse](#), string [numeroTel](#))
Crée une instance de la classe [Personne](#).
- void [Supprimer](#) ()
Méthode supprimant une personne de la liste des personnes.
- int [CompareTo](#) ([Personne](#) other)
- override string [ToString](#) ()

Fonctions membres publiques statiques

- static void [ImporterCSV< T >](#) (string nomFichier)
Permet d'importer les fichiers CSV ou TXT dont les données sont séparés par des ";".
- static List< T > [TrouverInstance< T >](#) (Predicate< T > predicate)
Permet de trouver toutes les instances de type T héritant de [Personne](#) répondant à la condition passée en argument.
- static [Personne](#) [InterfaceCreation](#) ()
On crée une personne en demandant les informations nécessaires à l'utilisateur, notamment le type de personne à créer.

Attributs protégés

- string [nom](#)
- string [prenom](#)
- string [adresse](#)
- string [numeroTel](#)

Attributs protégés statiques

- static List< [Personne](#) > [personnes](#) = new List<[Personne](#)>()

Propriétés

- static int [Count](#) [get]
- int [Identifiant](#) [get]
- string [Nom](#) [get, set]
"Les différentes informations peuvent être modifiables", selon l'énoncé. Ci dessous get et set permettant la modification, au besoin.
- string [Prenom](#) [get, set]
- string [Adresse](#) [get, set]
- string [NumeroTel](#) [get, set]

Fonctions membres privées

- delegate [Personne](#) [CreateurPersonne](#) ()
Permet d'encapsuler la méthode de création d'une personne.

Attributs privés

- int [identifiant](#)

Attributs privés statiques

- static int [dernierIdDonne](#) = 0

PersonneMorale

Fonctions membres publiques

- [PersonneMorale](#) (string [nom](#), string [adresse](#), string [numeroTel](#))
Crée une instance de la classe [Personne](#).

Attributs protégés

- int [identifiant](#)
- string [nom](#)
- string [adresse](#)

- string [numeroTel](#)

Propriétés

- string [Nom](#) [get, set]
"Les différentes informations peuvent être modifiables", selon l'énoncé. Ci dessous get et set permettant cela.
- string [Adresse](#) [get, set]
- string [NumeroTel](#) [get, set]
- int [Identifiant](#) [get]

Attributs privés statiques

- static int [dernierIdDonne](#) = 0

Recherche

Classe stockant nos delegates de fonctions de recherche pour un appel rapide

Fonctions membres publiques

- delegate List< T > [FonctionRecherche< T >](#) (string comparaison)
Fonction de recherche retournant les résultats de recherche selon une entrée fournie.

Template Parameters:

<i>T</i>	Type d'objet.
----------	---------------

Paramètres:

<i>comparaison</i>	Paramètre de recherche : valeur que le predicat recherchera dans une des informations de l'objet.
--------------------	---

Renvoie:

Liste de résultats

Fonctions membres publiques statiques

- static List< [Don](#) > [RechercheDonParStatutType< T >](#) (string statutCherche)
Permet de rechercher un don selon son statut et son type d'objet
- static List< [Don](#) > [RechercheDonParMoisType< T >](#) (string moisDuDon)
Permet de rechercher un don selon son mois de réception et son type d'objet
- static List< T > [RechercheParNomPersonneTypee< T >](#) (string nomCherche)
Trouve une personne par type et par nom
- static List< T > [RechercheParNumTelPersonneTypee< T >](#) (string numTelCherche)
Trouve une personne par type et par numéro de téléphone
- static List< T > [RechercheParNomLieuStockageTypee< T >](#) (string nomCherche)
Trouve une personne par type et par nom

Refrigerateur

Fonctions membres publiques

- [Refrigerateur](#) (double longueur, double largeur, double hauteur, double [prix](#))
- override string [ToString](#) ()

Fonctions membres publiques statiques

- static new [Materiel InterfaceCreation](#) ()
-

StockageAssociation

Fonctions membres publiques

- [StockageAssociation](#) (string [nom](#), string [adresse](#), string [numeroTel](#))
 - override string [ToString](#) ()
-

Table

Types publics

- enum [TypeTable](#) { [Cuisine](#), [Salon](#) }
- Enum contenant les types de table existants enum [FormeTable](#) { [Rectangulaire](#), [Carree](#), [Ronde](#) }

Enum contenant les formes de tables existantes Fonctions membres publiques

- [Table](#) ([TypeTable type](#), [FormeTable forme](#), double longueur, double largeur, double hauteur, double [prix](#)=0)
- override string [ToString](#) ()

Fonctions membres publiques statiques

- static new [Materiel InterfaceCreation](#) ()

Propriétés

- [FormeTable Forme](#) [get]
- [TypeTable Type](#) [get]

Attributs privés

- [TypeTable type](#)
 - [FormeTable forme](#)
-

Tri

Fonctions membres publiques

- delegate T [FonctionObtentionProprieteDon< T >](#) ([Don](#) don)
Fonction retournant pour un [Don](#) don une propriété de ce don sur laquelle on peut effectuer une comparaison.
- delegate T [FonctionObtentionProprieteDonLegue< T >](#) ([DonLegue](#) donLegue)

Fonctions membres publiques statiques

- static List< [Don](#) > [TriRefuse< T >](#) ([FonctionObtentionProprieteDon< T >](#) fonctionObtentionPropriete)
Fonction retournant une liste de dons refusés, triés selon la propriété dont on a le get en argument
- static List< [Don](#) > [TriAccepteStocke< T >](#) ([FonctionObtentionProprieteDon< T >](#) fonctionObtentionPropriete)

Fonction retournant une liste de dons acceptés ou stockés, triés selon la propriété dont on a le get en argument

- static List< [DonLegue](#) > [TriVenduDonne](#)< T > (FonctionObtentionProprieteDonLegue< T > fonctionObtentionPropriete)
Fonction retournant une liste de dons légués, triés selon la propriété dont on a le get en argument
- static List< [Don](#) > [TriLieuStockage](#)< M, T > ([LieuStockage](#) lieuStockage, FonctionObtentionProprieteDon< T > fonctionObtentionPropriete)
Fonction retournant une liste de dons stockés dont l'objet est de type M, triés selon la propriété dont on a le get en argument
- static void [sousMenuTriLieuStockage](#)< M, T > (string message, FonctionObtentionProprieteDon< T > fonctionObtentionPropriete)
Fonction demandant à l'utilisateur de choisir une instance de lieu de stockage, puis retournant une liste de dons stockés dedans dont l'objet est de type M, triés selon la propriété dont on a le get en argument
- static void [AfficherPrincipalesCategoriesEnStock](#) ()
Fonction affichant dans l'ordre les catégories les plus fréquentes des objets stockés.

Documentation des fonctions membres

static void AfficherPrincipalesCategoriesEnStock () [static]

Fonction affichant dans l'ordre les catégories les plus fréquentes des objets stockés.

delegate T FonctionObtentionProprieteDon< T > ([Don](#) don)

Fonction retournant pour un [Don](#) don une propriété de ce don sur laquelle on peut effectuer une comparaison.

Template Parameters:

<i>T</i>	Type de la propriété du don. Doit être comparable.
----------	--

Paramètres:

<i>don</i>	don dont on veut obtenir une propriété
------------	--

Renvoie:

Propriété d'un don

Contraintes de type

T : IComparable<T>

delegate T FonctionObtentionProprieteDonLegue< T > ([DonLegue](#) donLegue)

Contraintes de type

T : IComparable<T>

static void sousMenuTriLieuStockage< M, T > (string message, FonctionObtentionProprieteDon< T > fonctionObtentionPropriete) [static]

Fonction demandant à l'utilisateur de choisir une instance de lieu de stockage, puis retournant une liste de dons stockés dedans dont l'objet est de type M, triés selon la propriété dont on a le get en argument

Template Parameters:

<i>T</i>	Type de l'élément à comparer
<i>M</i>	Type de l'objet du don. Par exemple Materiel

Paramètres:

<i>message</i>	Message à afficher avant la liste d'objets
<i>fonctionObtentionPropriete</i>	

Contraintes de type

T : *Comparable*<*T*>
M : *Materiel*

static List<[Don](#)> TriAccepteStocke< T > (FonctionObtentionProprieteDon< T > fonctionObtentionPropriete)[static]

Fonction retournant une liste de dons acceptés ou stockés, triés selon la propriété dont on a le get en argument

```
TriAccepteStocke<int>(x => x.Identifiant);
```

Template Parameters:

<i>T</i>	Type de l'élément à comparer
----------	------------------------------

Paramètres:

<i>fonctionObtentionPropriete</i>	fonction donnant la propriété à comparer
-----------------------------------	--

Renvoie:

liste triée de dons acceptés ou stockés

Contraintes de type

T : *Comparable*<*T*>

static List<[Don](#)> TriLieuStockage< M, T > ([LieuStockage](#) lieuStockage, FonctionObtentionProprieteDon< T > fonctionObtentionPropriete)[static]

Fonction retournant une liste de dons stockés dont l'objet est de type M, triés selon la propriété dont on a le get en argument

Template Parameters:

<i>T</i>	Type de l'élément à comparer
<i>M</i>	Type de l'objet du don. Par exemple Materiel

Paramètres:

<i>fonctionObtentionPropriete</i>	fonction donnant la propriété à comparer
<i>lieuStockage</i>	Instance du lieu de stockage dans laquelle chercher

Renvoie:

liste triée de dons stockés

Contraintes de type

$T : IComparable<T>$

$M : Materiel$

**static List<[Don](#)> TriRefuse< T > (FonctionObtentionProprieteDon< T >
fonctionObtentionPropriete) [static]**

Fonction retournant une liste de dons refusés, triés selon la propriété dont on a le get en argument

Template Parameters:

T	Type de l'élément à comparer
-----	------------------------------

Paramètres:

<i>fonctionObtention Propriete</i>	fonction donnant la propriété à comparer
--	--

Renvoie:

liste triée de dons refusés

Contraintes de type

$T : IComparable<T>$

**static List<[DonLegue](#)> TriVenduDonne< T > (FonctionObtentionProprieteDonLegue< T
> fonctionObtentionPropriete) [static]**

Fonction retournant une liste de dons légués, triés selon la propriété dont on a le get en argument

Template Parameters:

T	Type de l'élément à comparer
-----	------------------------------

Paramètres:

<i>fonctionObtention Propriete</i>	fonction donnant la propriété à comparer
--	--

Renvoie:

liste triée de dons légués acceptés ou stockés

Contraintes de type

$T : IComparable<T>$

Vaisselle

Fonctions membres protégées

- [Vaisselle](#) (int [nombreDePieces](#), double [prix](#)=0)

Propriétés

- int [NombreDePieces](#) [get]

Attributs privés

- int [nombreDePieces](#)