

Table des matières

Interfaces	3
Placement	3
Conventions	4
Objets	5
Acteur_Externe : Public	5
Champs.....	5
Accesseurs/Mutateurs	5
Constructeurs	5
Méthodes	6
ActiveDirectory : Public Static	7
Champs.....	7
Accesseurs/Mutateurs	7
Méthodes	7
Type_Fonction : Public	8
Champs.....	8
Accesseurs/Mutateurs	8
Constructeurs	8
Méthodes	8
Salarie : Public	9
Champs.....	9
Accesseurs/Mutateurs	9
Constructeurs	9
Méthodes	10
Contact : Public	11
Champs.....	11
Accesseurs/Mutateurs	11
Constructeurs	11
BDD : Public.....	12
Champs.....	12
Accesseurs/Mutateurs	12
Constructeurs	12
Méthodes	12
Categorie : Public	21
Champs.....	21

Accesseurs/Mutateurs	21
Constructeurs	21
Méthodes	21
Commande : Public	22
Champs.....	22
Accesseurs/Mutateurs	22
Constructeurs	22
Constituer : Public	23
Champs.....	23
Accesseurs/Mutateurs	23
Constructeurs	23
Contenir : Public	24
Champs.....	24
Accesseurs/Mutateurs	24
Constructeurs	24
Produit : Public	25
Champs.....	25
Accesseurs/Mutateurs	25
Constructeurs	25
Service : Public	26
Champs.....	26
Accesseurs/Mutateurs	26
Constructeurs	26

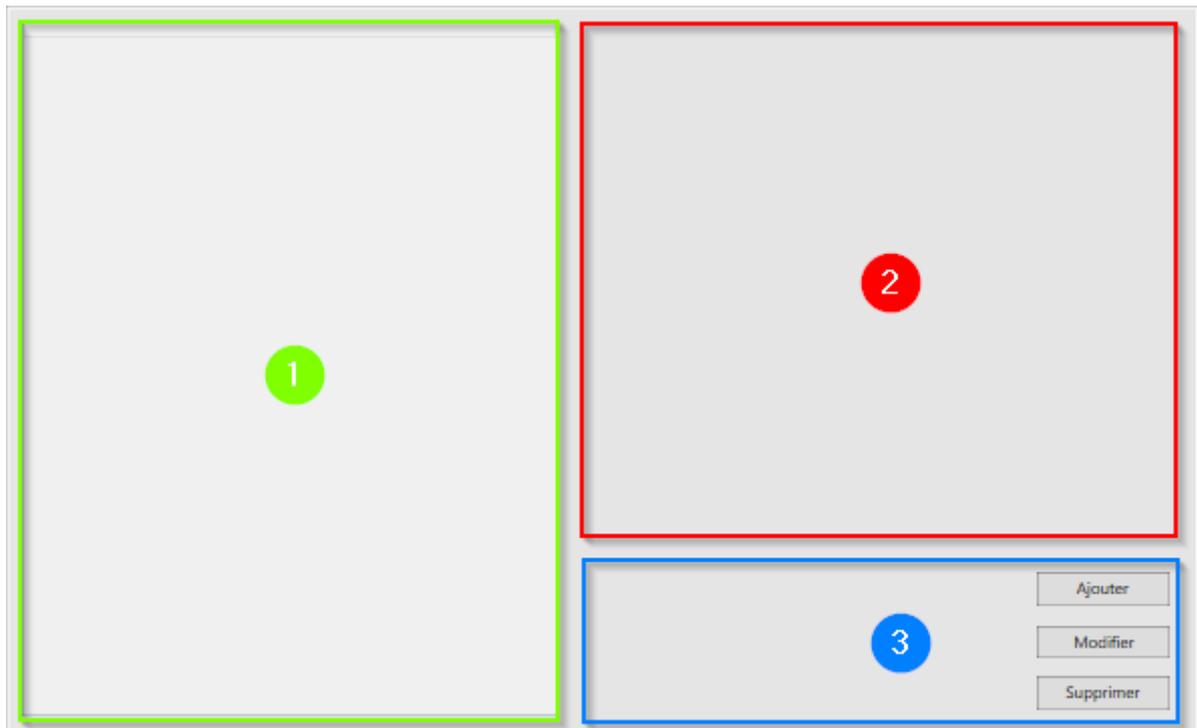
Cette documentation vous permettra dans les parties suivantes de savoir comment agencer vos nouveaux onglets, de savoir comment suivre les conventions adoptées mais aussi de savoir quel objet modifier en cas de besoin. Elle vous permet aussi de vous renseigner sur les différentes classes existantes afin de vous aider à les utiliser au mieux dans ce projet.

Interfaces

Placement

L'application est divisée en plusieurs onglets.

Les onglets qui seront principalement utilisés sont toujours construits de la manière suivante :



Une grille pour sélectionner des éléments (1) se situe à gauche de l'onglet car il s'agit du premier élément que l'utilisateur doit voir : la majorité des actions qu'il pourra effectuer nécessite d'abord une sélection.

Un formulaire constitué d'un nombre variable de champs (2) qui se remplissent automatiquement lors d'une sélection sur la grille (1). Il s'agit du deuxième élément qui doit être vu par l'utilisateur : la majorité des actions qu'il pourra effectuer nécessite de bien remplir ces champs.

La zone dédiée aux boutons d'action qui doit permettre à l'utilisateur de réaliser les actions qu'il souhaite, notamment ajouter, modifier ou supprimer. Il s'agit du dernier élément visible par l'utilisateur. Il est placé en bas à droite de l'écran pour éviter que l'utilisateur clique sur les boutons avant d'avoir sélectionné un élément dans la grille (1) et/ou d'avoir complété les champs (2). Cette zone est donc placée à la fin de l'onglet (dans le sens conventionnel de lecture).

Les derniers onglets (les moins fréquentés) ne sont pas construits de la même manière car ils ne proposent pas le même type d'informations.

Conventions

- Les noms des contrôles commencent toujours par une minuscule et par une abréviation du nom du contrôle.
- Les noms de variables commencent toujours par une minuscule.
- Les noms des champs commencent toujours par un underscore (_).
- Les noms des accesseurs/mutateurs commencent toujours par une majuscule.
- Les noms des constructeurs commencent toujours par une majuscule.
- Les noms des méthodes commencent toujours par une majuscule.
- L'opérateur ternaire doit être utilisé dès que possible (`expr1 ? expr2 : expr3`).
- Toutes les classes publiques non statiques ont toujours un constructeur vide en cas de besoin.

Une classe passerelle a été créée (BDD.cs) pour regrouper les méthodes permettant d'interagir avec la base de données dans le même fichier. Si une table est ajoutée dans la base de données, il suffit d'ajouter au minimum une fonction pour récupérer toutes les lignes de cette table, une autre pour modifier ou supprimer une ligne et enfin une dernière pour l'ajout.

Une classe a été créée (ActiveDirectory.cs) pour permettre de récupérer les informations d'un utilisateur plus facilement. Une instance de Salarie nommé « `connectedUser` » contient des informations sur le salarié après la connexion grâce à la classe ActiveDirectory. Un court descriptif de cette classe est disponible dans cette documentation.

Objets

L'objectif de cette section est de regrouper toutes les informations sur chaque classe. On trouve pour chacune la liste des champs, les accesseurs/mutateurs à utiliser en dehors des classes elles-mêmes, les constructeurs ainsi que les méthodes s'il y en a.

Une description de la valeur de sortie est donnée avec chaque méthode.

Acteur Externe : Public

Champs

Protected	Int	_idExt
Protected	String	_nomExt
Protected	String	_prenomExt
Protected	String	_telExt
Protected	String	_mailExt
Protected	String	_rueExt
Protected	String	_villeExt
Protected	String	_cpExt
Protected	Bool	_estClient
Protected	Bool	_estEntreprise

Accesseurs/Mutateurs

Public	Int	IdExt
Public	String	NomExt
Public	String	PrenomExt
Public	String	TelExt
Public	String	MailExt
Public	String	RueExt
Public	String	VilleExt
Public	String	CpExt
Public	Bool	EstClient
Public	Bool	EstEntreprise

Constructeurs

```
Public
Signature      Acteur_Externe()
                -- Vide --
```

```
Public
Signature      Acteur_Externe()
                String      nom
                String      prenom
                String      tel
                String      mail
                String      rue
                String      ville
```

String	cp
Bool	estClient
Bool	estEntreprise

Public
Signature

Acteur_Externe()	
Int	id
String	nom
String	prenom
String	tel
String	mail
String	rue
String	ville
String	cp
Bool	estClient
Bool	estEntreprise

Ce constructeur est utilisé lors de la récupération de tous les acteurs externes afin de conserver leur identifiant (qui sera utiliser plus tard, pour une modification ou une suppression).

Méthodes

Public Override
Paramètres
Sortie

String

ToString()
-- Vide --

Une chaîne de caractères constituée du nom en majuscule et du prénom de l'acteur externe.

ActiveDirectory : Public Static

Champs

Private Static	String	_server
Private Static	String	_user
Private Static	String	_password
Private Static	DirectoryEntry	_connection

Accesseurs/Mutateurs

Public Static	String	Server
Public Static	String	User
Public Static	String	Password
Public Static	DirectoryEntry	Connection

Méthodes

Public Static	DirectoryEntry	Connect()
Paramètres		-- Vide --

Sortie Retourne une instance de DirectoryEntry. Cette méthode permet une connexion à l'ActiveDirectory (Server) avec les identifiants User et Password.

Public Static	Salarie	getUser()
Paramètres		String username

Sortie Retourne une instance de Salarie. Cette méthode permet la recherche d'un utilisateur 'username' sur l'ActiveDirectory. Null est retourné si aucun utilisateur n'est trouvé. Sinon, les informations de l'utilisateur forment une instance de Salarie. Celle-ci sera alors retournée en cas de succès de la recherche.

Type_Fonction : Public

Champs

Private	Int	_idTypeFonction
Private	String	_libTypeFonction

Accesseurs/Mutateurs

Public	Int	IdFonction
Public	String	LibTypeFonction

Constructeurs

Public	Type_Fonction()
Signature	-- Vide --

Public	Type_Fonction()	
Signature	Int	idTypeFonction
	String	libTypeFonction

Ce constructeur est appelé plusieurs fois pour remplir une liste de Type_Fonction retournée par la méthode Select_Type_Fonctions() de la classe BDD.

Méthodes

Public Override	String	Tostring()
Paramètres		-- Vide --

Sortie	Une chaîne de caractères qui contient le libellé du type de contact. Elle est utilisé pour afficher le moyen de contact sur l'interface (dans un ComboBox nommé « cboMoyenContact »).
--------	---

Salarie : Public

Champs

Private	Int	_idSalarie
Private	Type_Fonction	_fonction
Private	String	_nomSalarie
Private	String	_prenomSalarie
Private	String	_telSalarie
Private	String	_adrMailSalarie
Private	String	_loginSalarie
Private	String	_passSalarie
Private	String	_rueSalarie
Private	String	_villeSalarie
Private	String	_cpSalarie
Private	List<String>	_groupesSalarie

Accesseurs/Mutateurs

Public	Int	IdSalarie
Public	Type_Fonction	Fonction
Public	String	NomSalarie
Public	String	PrenomSalarie
Public	String	TelSalarie
Public	String	AdrMailSalarie
Public	String	LoginSalarie
Public	String	PassSalarie
Public	String	RueSalarie
Public	String	VilleSalarie
Public	String	CpSalarie
Public	List<String>	GroupesSalarie

Constructeurs

Public	Salarie()
Signature	-- Vide --

Public	Salarie()	
Signature	Int	idSal
	Type_Fonction	fonction
	String	nom
	String	prenom
	String	tel
	String	mail
	String	login
	String	pass
	String	rue
	String	ville
	String	cp

Ce constructeur est utilisé dans la classe BDD notamment pour sélectionner tous les salariés ou sélectionner le salarié qui a pris contact avec un acteur externe.

```
Public Signature Salarie()  
String nom  
String prenom  
String tel  
String mail  
String login  
String rue  
String ville  
String cp  
String grps
```

Ce constructeur est utilisé dans la classe ActiveDirectory pour récupérer l'utilisateur qui tente de se connecter (dans getUser()).

Méthodes

```
Public Override String ToString()  
Paramètres -- Vide --  
Sortie Une chaîne de caractères constituée du nom en  
majuscule et du prénom du salarié.
```

Contact : Public

Cette classe correspond à un contact entre un salarié et un acteur externe à une date précise.

Champs

Private	Salarie	_salarie
Private	Acteur_Externe	_acteurExterne
Private	Int	_idContact
Private	String	_etatContact
Private	DateTime	_dateContact
Private	String	_recapContact
Private	String	_typeContact

Accesseurs/Mutateurs

Public	Salarie	Salarie
Public	Acteur_Externe	ActeurExterne
Public	Int	IdContact
Public	String	EtatContact
Public	DateTime	DateContact
Public	String	RecapContact
Public	String	TypeContact

Constructeurs

Public	Signature	Contact()
		-- Vide --

Public	Signature	Contact()
		Salarie sal
		Acteur_Externe ext
		Int idContact
		String etat
		DateTime date
		String recap
		String typeContact

Ce constructeur est utilisé dans la classe BDD pour récupérer tous les contacts d'un même salarié (dans la méthode `Select_User_Contacts()`). Il est aussi utilisé pour récupérer une instance de `Contact` lors de la sélection d'un contact par exemple.

BDD : Public

La classe BDD correspond à la classe passerelle qui permet d'interagir avec la base de données grâce à toutes une série de méthodes ou des requêtes SQL sont effectuées pour obtenir toutes les occurrences d'une table, d'en ajouter, d'en supprimer ou d'en modifier une par exemple.

Champs

Private	MySQLConnection	_connection
Private	String	_ipServer
Private	String	_dataBaseName
Private	String	_uid
Private	String	_password
Private	Int	_nbAlerts

Accesseurs/Mutateurs

Public	MySQLConnection	Connection
Public	String	IpServer
Public	String	DataBaseName
Public	String	UID
Public	String	Password
Public	Int	NbAlerts

Constructeurs

Public
Signature
BDD()
-- Vide --

Public
Signature
BDD()
String ip
String dbname
String uid
String pass

Ce constructeur permet de créer une instance de MySQLConnection. La connexion sera ouverte ou fermée via les méthodes Connect() et Disconnect().

Méthodes

Public Paramètres	Bool	Connect() <i>-- Vide --</i>
Sortie	Tente d'ouvrir la connexion. Retourne true en cas de succès sinon une erreur est affichée et la méthode retourne false.	
Public Paramètres	Bool	Disconnect() <i>-- Vide --</i>
Sortie	Tente de fermer la connexion. Retourne false en cas de succès sinon true.	

Le booléen retourner par ces deux fonctions représentent en réalité l'état de la connexion à la base de données.

Public Paramètres	Void	Insert_Commande() Int idExt String dateCde
Fonction	Ajoute une commande dans la table « Commande » de la base de données faite par un acteur externe à une date précise.	
Public Paramètres	Void	Insert_Constituer_Service() Int idServ Int idCde
Fonction	Ajoute un service à une commande existante dans la table « Constituer ».	
Public Paramètres	Void	Insert_Contenir_Produit() Int idProd Int idCde Int qte
Fonction	Ajoute un ou plusieurs produits à une commande existante dans la table « Contenir ».	
Public Paramètres	Void	Insert_Produit() String libProd String descProd Double prixProd String photoProd Int idCat
Fonction	Ajoute un produit dans la table « Produit ».	
Public Paramètres	Void	Insert_Service() String libServ String descServ Double prixServ String photoServ Int idCat
Fonction	Ajoute un service dans la table « Service ».	

Public Paramètres	Void	Insert_Acteur_Externe() String nomExt String prenomExt String telExt String mailExt String rueExt String villeExt String cpExt Bool estClient Bool estEntreprise
Fonction	Ajoute un acteur externe dans la table « acteur_externe ».	
Public Paramètres	Void	Insert_Salarie() Int idTypeFonction String nomSalarie String prenomSalarie String telSalarie String mailSalarie String loginSalarie String passSalarie String rueSalarie String villeSalarie String cpSalarie
Fonction	Ajoute un salarié dans la table « salarie ».	
Public Paramètres	Void	Insert_Contact() Int idSalarie Int idExt String etatContact String dateContact String recapContact String typeContact
Fonction	Ajoute un contact dans la table « contact ».	
Public Paramètres	Void	Insert_Categorie() Int idCat String libCat
Fonction	Ajoute une catégorie de produit ou de service dans la table « categorie ».	
Public Paramètres	Void	Insert_Type_Fonction() Int idTypeFonction String libTypeFonction
Fonction	Ajoute une fonction de salarié dans la table « type_fonction ».	

Public Paramètres	Void	Update_Acteur_Externe() Int id String nom String prenom String tel String mail String rue String ville String cp Bool estClient Bool estEntreprise
----------------------	------	--

Fonction Modifie un acteur externe portant l'id « id » dans la base de données. Les autres paramètres représentent les nouvelles données.

Public Paramètres	Void	Update_Salarie() Int idSalarie Int idTypeFonction MySQLConnection dateCde
----------------------	------	--

Fonction Modifie un salarié portant l'id « idSalarie » dans la table « salarie ». Les autres paramètres représentent les nouvelles données.

Public Paramètres	Void	Update_Commande() Int idCde Int idExt String nom String prenom String tel String mail String login String pass String rue String ville String cp
----------------------	------	---

Fonction Modifie une commande portant l'id « idCde » faite par un acteur externe portant l'id « idExt » dans la table « commande ». Les autres paramètres représentent les nouvelles données.

Public Paramètres	Void	Update_Constituer_Service() Int idServWhere Int idExtWhere Int idCdeWhere Int idServ Int idExt Int idCde
----------------------	------	--

Fonction	Permet de modifier un service dans une commande en remplaçant son identifiant ou celui de la commande dans laquelle il se trouve, ou bien l'identifiant de celui qui a commandé le service par de nouveaux identifiants dans la table « constituer ».
----------	---

Public Paramètres	Void	Update_Contenir_Produit() Int idProdWhere Int idExtWhere Int idCdeWhere Int idProd Int idExt Int idCde Int Quantite
----------------------	------	--

Fonction	Permet de modifier un produit dans une commande en remplaçant son identifiant ou celui de la commande dans laquelle il se trouve, ou bien l'identifiant de celui qui a commandé le service par de nouveaux identifiants dans la table « contenir ». Il est aussi possible de modifier la quantité de produit présent dans la commande.
----------	--

Public Paramètres	Void	Update_Contact() Int idSalarie Int idExt Int idContact String etatContact String dateContact String recapContact String typeContact
----------------------	------	--

Fonction	Permet de modifier un contact n° « idContact » entre un acteur externe (« idExt ») et un salarié (« idSalarie ») dans la table « contact ». Les autres paramètres correspondent aux nouvelles données du contact.
----------	---

Public Paramètres	Void	Update_Produit() Int idProd String libProd String descProd Double prixProd String photoProd Int idCat
Fonction	Permet de modifier un produit (« idProd ») dans la table « produit ». Les autres paramètres correspondent aux nouvelles données du produit.	
Public Paramètres	Void	Update_Service() Int idServ String libServ String descServ Double prixServ String photoServ Int idCat
Fonction	Permet de modifier un service (« idServ ») dans la table « service ». Les autres paramètres correspondent aux nouvelles données du service.	
Public Paramètres	Void	Update_Type_Fonction() Int idTypeFonction String libTypeFonction
Fonction	Permet de modifier une fonction de salarié dans la table « type_fonction ». « libTypeFonction » sera la valeur modifiée.	
Public Paramètres	Void	Update_Categorie() Int idCat String libCat
Fonction	Permet de modifier une catégorie de produit ou de service dans la table « categorie ». « libCat » sera la valeur modifiée.	
Public Paramètres	Void	Delete() Int[] num String nomTable String[] nomColonne
Fonction	<p>Permet de supprimer toutes les lignes portant les identifiants (du paramètre « num ») pour les colonnes « nomColonne » dans la table « nomTable ».</p> <p><u>Exemple :</u> Delete(new int[] {1,3,2}, 'contenir', new string[] {'idProd','idExt','idCde'}); Aura pour effet de supprimer le produit n°1 dans la commande n°2 faite par l'acteur n°3 dans la table « contenir ».</p>	

Public Paramètres	List<MySQLConnection>	Select_Acteurs_Externes() -- Vide --
Sortie	Retourne tous les acteurs externes de la base de données sous forme de liste.	
Public Paramètres	List<Salarie>	Select_Salaries() -- Vide --
Sortie	Retourne tous les salariés de la base de données sous forme de liste.	
Public Paramètres	List<Commande>	Select_Commandes() -- Vide --
Sortie	Retourne toutes les commandes de la base de données sous forme de liste.	
Public Paramètres	List<Constituer>	Select_Constituer_Services() -- Vide --
Sortie	Retourne toutes les services contenus dans des commandes de la base de données sous forme de liste.	
Public Paramètres	List<Contenir>	Select_Contenir_Produits() -- Vide --
Sortie	Retourne toutes les produits contenus dans des commandes de la base de données sous forme de liste.	
Public Paramètres	List<Contact>	Select_All_Contacts() -- Vide --
Sortie	Retourne toutes les contacts entre acteurs externes et les salariés sans exceptions.	
Public Paramètres	List<Contact>	Select_User_Contacts() Salarie sal
Sortie	Retourne toutes les contacts entre acteurs externes et le salarié qui utilise l'application.	
Public Paramètres	List<Produit>	Select_Produits() -- Vide --
Sortie	Retourne tous les produits de la base de données sous forme de liste.	
Public Paramètres	List<Service>	Select_Services() -- Vide --
Sortie	Retourne tous les services de la base de données sous forme de liste.	

Public Paramètres	List<Type_Fonction>	Select_Type_Fonction() -- Vide --
Sortie	Retourne toutes les fonctions de salariés de la base de données sous forme de liste.	
Public Paramètres	List<Categorie>	Select_Categories() -- Vide --
Sortie	Retourne toutes les catégories de produits et services de la base de données sous forme de liste.	
Public Paramètres	Double	Count_Contact() -- Vide --
Sortie	Retourne le nombre de contacts entre acteurs externes et salariés au total dans la base de données.	
Public Paramètres	Double	Count_Contact() -- Vide --
Sortie	Retourne le nombre de contacts entre acteurs externes et salariés au total dans la base de données.	
Public Paramètres	Double	Count_RDV_AFaire() -- Vide --
Sortie	Retourne le nombre de contacts qui ne sont pas terminés dans la base de données.	
Public Paramètres	Double	Count_RDV_Fait() -- Vide --
Sortie	Retourne le nombre de contacts qui sont terminés dans la base de données.	
Public Paramètres	Double	Count_Clients_Et_Prospects() -- Vide --
Sortie	Retourne le nombre total d'acteurs externes.	
Public Paramètres	Double	Count_Clients() -- Vide --
Sortie	Retourne le nombre total d'acteurs externes qui sont des clients.	
Public Paramètres	Double	Count_Entreprises() -- Vide --
Sortie	Retourne le nombre total d'acteurs externes qui sont des entreprises.	

Public Paramètres	Double	Count_Contacts_Tel() <i>-- Vide --</i>
Sortie	Retourne le nombre total de contacts par téléphone.	

Public Paramètres	Double	Count_Contacts_Mail() <i>-- Vide --</i>
Sortie	Retourne le nombre total de contacts par mail.	

Public Paramètres	Double	Count_Contacts_Courrier() <i>-- Vide --</i>
Sortie	Retourne le nombre total de contacts par courrier.	

Public Paramètres	Double	Count_Contacts_Rendez_Vous() <i>-- Vide --</i>
Sortie	Retourne le nombre total de contacts par rendez-vous.	

Public Paramètres	Double	Count_Contacts_Autres() <i>-- Vide --</i>
Sortie	Retourne le nombre total de contacts qui ne sont pas par téléphone, mail, courrier ou rendez-vous.	

Categorie : Public

Champs

Private

Private

Int

String

_idCat

_libCat

Accesseurs/Mutateurs

Public

Public

Int

String

IdCat

LibCat

Constructeurs

Public

Signature

Categorie()

-- Vide --

Public

Signature

Categorie()

Int

String

idCat

libCat

Ce constructeur est utilisé dans la classe BDD pour sélectionner toutes les catégories ou bien pour la sélection de tous les produits.

Méthodes

Public Override

Paramètres

Sortie

String

ToString()

-- Vide --

Retourne le libellé de catégorie.

Commande : Public

Champs

```
Private      Int      _idCde
Private      Int      _idExt
Private      DateTime _dateCde
```

Accesseurs/Mutateurs

```
Public      Int      IdCde
Public      Int      IdExt
Public      DateTime DateCde
```

Constructeurs

```
Public      Commande()
Signature   -- Vide --
```

```
Public      Commande()
Signature   Int      idCommande
            Int      idActeur
            DateTime dateCom
```

Ce constructeur est utilisé pour la sélection de toutes les commandes dans la classe BDD.

Constituer : Public

Un Objet Constituer correspond à un service demandé par un acteur externe dans une commande.

Champs

Private	Int	_idServ
Private	Int	_idExt
Private	Int	_idCde

Accesseurs/Mutateurs

Public	Int	IdServ
Public	Int	IdExt
Public	Int	IdCde

Constructeurs

Public		Constituer()
Signature		-- Vide --

Public		Constituer()	
Signature	Int		idServ
	Int		idExt
	Int		idCde

Ce constructeur est utilisé pour la sélection de tous les services commandés dans toutes les commandes faites par n'importe quel acteur externe.

Contenir : Public

Un Objet Contenir correspond à un produit (ou plusieurs, selon la quantité) demandé par un acteur externe dans une commande.

Champs

Private	Int	_idProd
Private	Int	_idExt
Private	Int	_idCde
Private	Int	_quantite

Accesseurs/Mutateurs

Public	Int	IdProd
Public	Int	IdExt
Public	Int	IdCde
Public	Int	Quantite

Constructeurs

Public		Contenir()
Signature		-- Vide --

Public		Contenir()	
Signature	Int		idProd
	Int		idExt
	Int		idCde
	Int		qte

Ce constructeur est utilisé pour la sélection de tous les produits commandés dans toutes les commandes faites par n'importe quel acteur externe.

Produit : Public

Champs

Private	Int	_idProd
Private	String	_libProd
Private	String	_descProd
Private	Double	_prixProd
Private	String	_photoProd
Private	Categorie	_categorie

Accesseurs/Mutateurs

Public	Int	IdProd
Public	String	LibProd
Public	String	DescProd
Public	Double	PrixProd
Public	String	PhotoProd
Public	Categorie	Categorie

Constructeurs

Public	Produit()
Signature	-- Vide --

Public	Produit()
Signature	Int idProd
	String lib
	String desc
	Double prix
	String photo
	Categorie cat

Ce constructeur est notamment utilisé dans la classe BDD pour construire la liste de produits retournés par la fonction `Select_Produits()`.

Service : Public

Champs

Private	Int	_idServ
Private	String	_libServ
Private	String	_descServ
Private	Double	_prixServ
Private	String	_photoServ
Private	Int	_idCat

Accesseurs/Mutateurs

Public	Int	IdServ
Public	String	LibServ
Public	String	DescServ
Public	Double	PrixServ
Public	String	PhotoServ
Public	Int	IdCat

Constructeurs

Public	Service()
Signature	-- Vide --

Public	Service()	
Signature	Int	idServ
	String	lib
	String	desc
	Double	prix
	String	photo
	Int	idCat

Ce constructeur est notamment utilisé dans la classe BDD pour construire la liste de produits retournés par la fonction `Select_Services()`.