---------- Propriétés d'une classe ----------

Les propriétés d'une classe seront déclarées ainsi :

1) Le type de la propriété par sa première lettre
2) Underscore
3) Nom de la variable adéquat à la BDD (Commençant par une majuscule)

Ex :

Id de type int → i_Id;
Nom du client de type String → s_NomClient;
Valeur d'un solde en Float → f_Solde;
Une date de type date → dt_DateRetrait;
// (seule variable avec 2 lettres, pour pas confondre avec le type Double)
Une propriétés de type objet → o_Compte;
Une liste d'objet → a_LesComptes;
// (déterminant « les » pour dissocier l'objet à la liste)

---------- Variable internes et paramètre d'une fonction ou procédure ----------

Les variables d'une méthode seront déclarées suivant ces instructions :

1) Le type de la variable sera écrite en 3 lettres
2) Nom de la variable, en cohérence avec la BDD, commençant par une majuscule

Ex :

Id de type int → intId;
Nom du client de type String → strNomClient;
Valeur d'un solde en Float → fltSolde;
Une date de type date → dateDateRetrait;
Une propriétés de type objet → objCompte;
Une liste d'objet → arrObjComptes;
// (« Obj » au lieu de « les » pour comprendre que c'est une liste d'objet)

---------- Déclaration des Getter / Setter ----------

Les méthodes Getter et Setter seront déclarer sous cette forme :

Getter :

```
public int getId() {
    return this.i_Id;
```

}

Setter :

```
public void setId(int intId) {
        this.i_Id = intId;
}
```

---------- Signature d'une méthode ----------

Les signatures des méthodes seront déclarées sous cette forme :

```
public Client (int intIdClient, String strNomClient, Float fltSolde, date dateDateRetrait, Compte objCompte, array<Compte> arrObjComptes) {

        …..
}
```

---------- Nomination des fichiers ----------

L'application utiliseras l'architecture en couche (BO, DAL, BLL et IHM).
On aura donc un dossier pour chaque couche :

- BO : classe métiers
- DAL : Objet et méthode de dialogue entre la couche BO et base de données
- BLL : Tous les controller (classe qui commande l'affichage et l'exécution)
- IHM : les interfaces homme-machine

Par conséquent, les classes devrons être nommer ainsi :

1) En minuscule : l'intitulé de la couche
2) Nom de la table qui est lié / L'action censé faire
3) Le 2) commence par une majuscule

Exemple :

Un controller qui gère l'affichage des comptes en banque → bllAfficherComptes

Une classe métier de la table Compte → boCompte

Une interface qui affiche les comptes → ihmComptes

Si il s'agit d'un affichage multiple d'une données, les classes doivent être au pluriel

---------- ATTENTION ----------

1) **Attention au pointage ! On sélectionne la propriétés de la classe (this.i_Id) pour le pointer vers la variable locale (intId) !**
2) **Attention aux espacements !**
3) **Attention à l'indentation !**
4) **Attention à l'ordre de priorité lors des déclarations de paramètres ! On suit l'ordre des colonnes dans la table de la DB !**