



# POO : LES CLASSES COMPOSES D'OBJETS

Julien Condomines

SIO 1



## TABLE DES MATIERES

Introduction .....	3
Contexte d'exemple « Livre » .....	4
Développement des constructeurs .....	5
Diagramme UML de la classe livre .....	6
Diagramme UML de la classe Auteur .....	7
Diagramme UML de la classe Maisonédition.....	8
Autre contexte d'exemple « Prod'Huit » .....	9
Développement des constructeurs .....	10
La classe Date .....	11
Diagramme UML de la classe Livraison .....	12
Diagramme UML de la classe Categorie .....	13
Diagramme UML de la classe client .....	14
Diagramme UML de la classe Date .....	15
Diagramme UML de la classe Produits .....	16
Diagramme UML de la classe Livreur.....	17
Conclusion .....	18
Annexe.....	19
Classe Livre.....	19
Classe Auteur .....	20
Classe MaisonEdition .....	21
Classe Livraisons .....	22

## Les classes composé d'objets

Classe Produits .....	23
Classe Date .....	25
Classe Client .....	26
Classe Livreur.....	27
Classe Catégorie .....	28

## INTRODUCTION

Comme nous l'avons précédemment, la programmation orientée objet consiste à concevoir une application sous la forme de « briques » logicielles appelées des objets. Chaque objets jouent un rôle précis et peut communiquer avec les autres objets. Les interactions entre les différents objets vont permettre à l'application de réaliser ce qu'on lui demande.

Nous savons qu'un objet est une entité qui modélise un élément du domaine étudié : un employé, un compte bancaire, une facture, etc. Un objet est toujours créé d'après un modèle, qui est appelé sa classe.

Sauf dans les cas les plus simples, on ne pourra pas modéliser fidèlement le domaine étudié en se contentant de concevoir une seule classe. Il faudra définir plusieurs classes et donc instancier des objets de classes différentes. Cependant, ces objets doivent être mis en relation afin de pouvoir communiquer.

## CONTEXTE D'EXEMPLE « LIVRE »

On cherche dans un premier temps ici à créer une classe [Livre](#), dans laquelle on fera figurer les données représentant un livre, comme son numéro, son titre, ainsi qu'un compteur pour le nombre de livre que l'on crée.

Suite à cela, on cherche à donner plus d'information sur ce livre comme son auteur. Cependant, c'est une information qui peut être communes à plusieurs livres, nous allons donc l'instancier dans une classe différente appelés « [Auteur](#) » afin de pouvoir l'utiliser dès que nous en avons besoins.

Pour aller encore plus loin, nous cherchons à attribuer une maison d'édition aux auteurs, que nous allons la aussi instancier dans une classe différentes appelés « [MaisonEdition](#) ».

Cependant, comment faire communiquer ces différentes classes entre-elles ?

## DEVELOPPEMENT DES CONSTRUCTEURS

Afin de créer ces différents auteur et ces différentes maisons d'édition, il est nécessaire de, dans chacune de leurs classes respective, créer un constructeur Ref(Annexe).

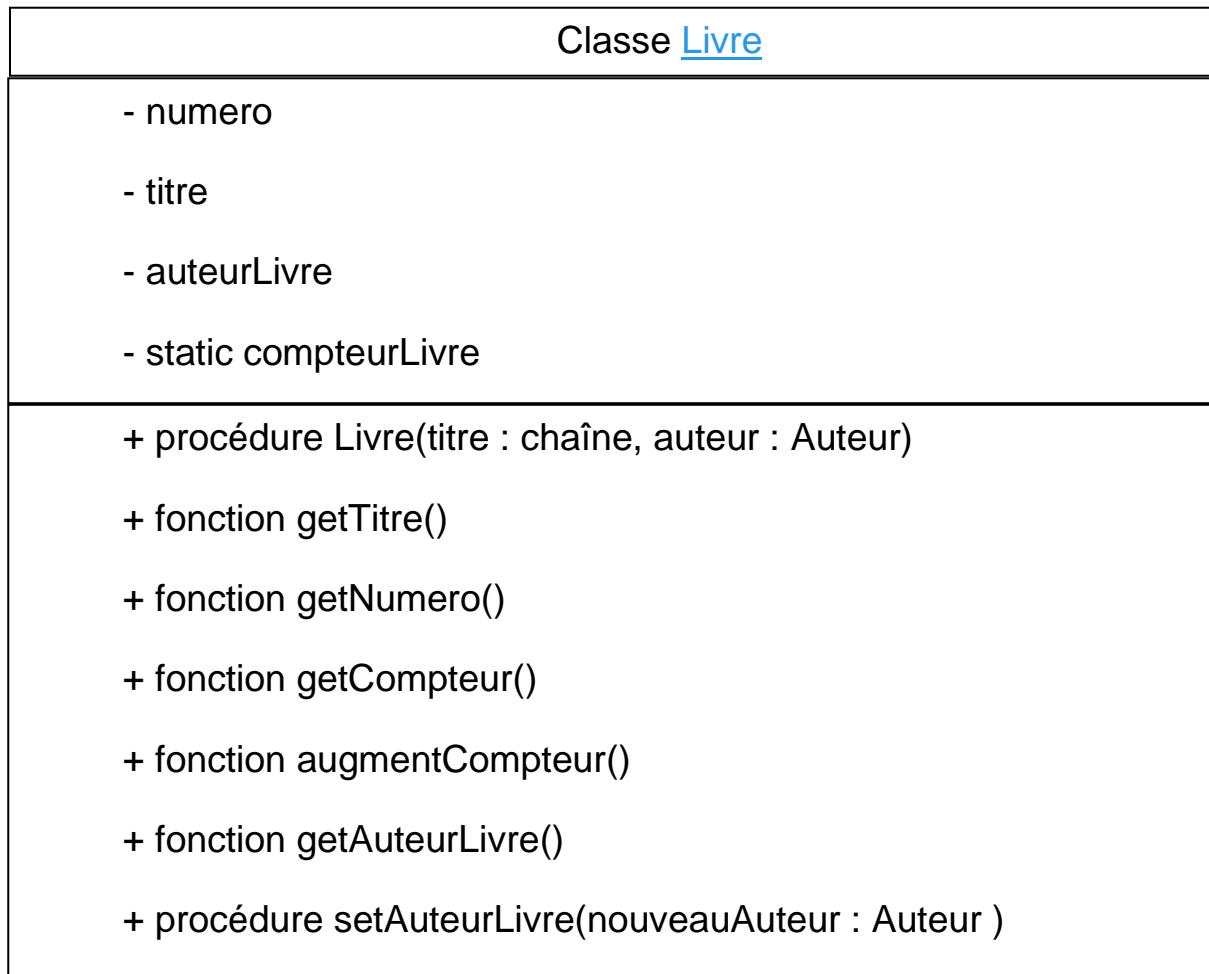
Cependant ce n'est pas suffisants, les constructeurs ne sont pas connectés entre eux, on va donc rajouter dans le constructeur de la classe [Auteur](#) un argument maisonEdition avec pour type la classe [MaisonEdition](#).

```
1 référence
public Auteur(string nom, string prenom, MaisonEdition maisonEdition )
{
    this.nom = nom;
    this.prenom = prenom;
    nomMaisonEdition = maisonEdition;
}
```

De même pour le constructeur de la classe [Livre](#) auquel on va ajouter un argument auteur de type [Auteur](#) pour faire référence à sa classe.

```
1 référence
public Livre (string titre, Auteur auteur)
{
    this.titre = titre;
    augmentCompteur();
    this.numero = Livre.getCompteur();
    auteurLivre = auteur;
}
```

## DIAGRAMME UML DE LA CLASSE LIVRE



## DIAGRAMME UML DE LA CLASSE AUTEUR

Classe <u>Auteur</u>
<ul style="list-style-type: none"><li>- nom</li><li>- prenom</li><li>- nomMaisonEdition</li></ul>
<ul style="list-style-type: none"><li>+ procédure Auteur(nom : chaîne, prenom : chaîne, maisonEdition : MaisonEdition)</li><li>+ fonction getNom()</li><li>+ fonction getPrenom()</li><li>+ fonction getMaisonEdition()</li><li>+ procédure setNomMaisonEdition (nouveauNomME : MaisonEdition )</li><li>+ procédure setPrenom (nouveauPrenom : string )</li><li>+ procédure setNom (nouveauNom : string )</li></ul>



## DIAGRAMME UML DE LA CLASSE MAISONEDITION

Classe <a href="#">MaisonEdition</a>
<ul style="list-style-type: none"><li>- nom</li><li>- cotation</li></ul>
<ul style="list-style-type: none"><li>+ procédure MaisonEdition(nom : chaîne, cotation : entier)</li><li>+ fonction getNom()</li><li>+ fonction getCotation()</li><li>+ procédure setCotation (nouveauCotation : entier )</li><li>+ procédure setNom (nouveauNom : string )</li></ul>

## AUTRE CONTEXTE D'EXEMPLE « PROD'HUIT »

On enchaîne avec un cas pratique plus concret, qui va nous permettre de mieux appréhender l'étendue de la commande que nous avons découvert auparavant, nous allons ici faire communiquer un nombre de classes bien plus important.

On prendra le cas Prod'Huit, composé de 6 classes différentes, permettant de mettre en place les processus de suivi de livraisons d'un produit. Toutes les classes seront reliés directement ou non à la classe « Livraison » qui fera office de classe principal.

Nous allons aussi découvrir les méthodes dates avec notamment la commande « DateTime » qui permet d'initialiser une variable avec des données de temp.

## DEVELOPPEMENT DES CONSTRUCTEURS

Nous avons tout d'abord la classe [Produit](#) qui est reliée aux classes [Catégorie](#) et [Date](#) par les variables respectives `categProduit` et `datePeremption`, ce qui va permettre d'établir la catégorie du produit à livrer ainsi que sa date de péremption, qui peuvent toutes deux varier d'un produit à un autre, ce pourquoi elles doivent être séparées de la classe `Produit`.

```
0 références
public Produits(string referenceSai, string nomProduitSai, double coutEstimekmSai, Date datePreemptionSai)
{
    this.reference = referenceSai;
    this.nomProduit = nomProduitSai;
    this.coutEstimeKm = coutEstimekmSai;
    this.datePeremption = datePreemptionSai;
}
```

Ensuite on crée la classe [Livraison](#), qui va être reliée à toutes les autres classes, afin de permettre de voir la livraison d'un produit. Il y a tout d'abord la classe [Date](#) avec la `dateDeLivraison` qui permet de savoir quand est-ce que le produit doit être livré, ensuite il y a la classe [Produit](#) avec la variable `produitLivre` qui permet de savoir le produit qui va être livré, il y a aussi la classe [Client](#) avec la variable `clientLivre` qui permet de connaître le nom du client qui va être livré ainsi que son adresse, enfin il y a la classe [Livreur](#) avec la variable `livreurProduit` qui permet d'obtenir les informations sur le livreur qui va effectuer cette livraison.

```
0 références
public Livraison (int numero, Date dateLivraison, Produit produitLivre, Client clientLivre, Livreur livreurProduit)
{
    this.numero = numero;
    this.dateLivraison = dateLivraison;
    this.produitLivre = produitLivre;
    this.clientLivre = clientLivre;
    this.livreurProduit = livreurProduit;
}
```

## LA CLASSE DATE

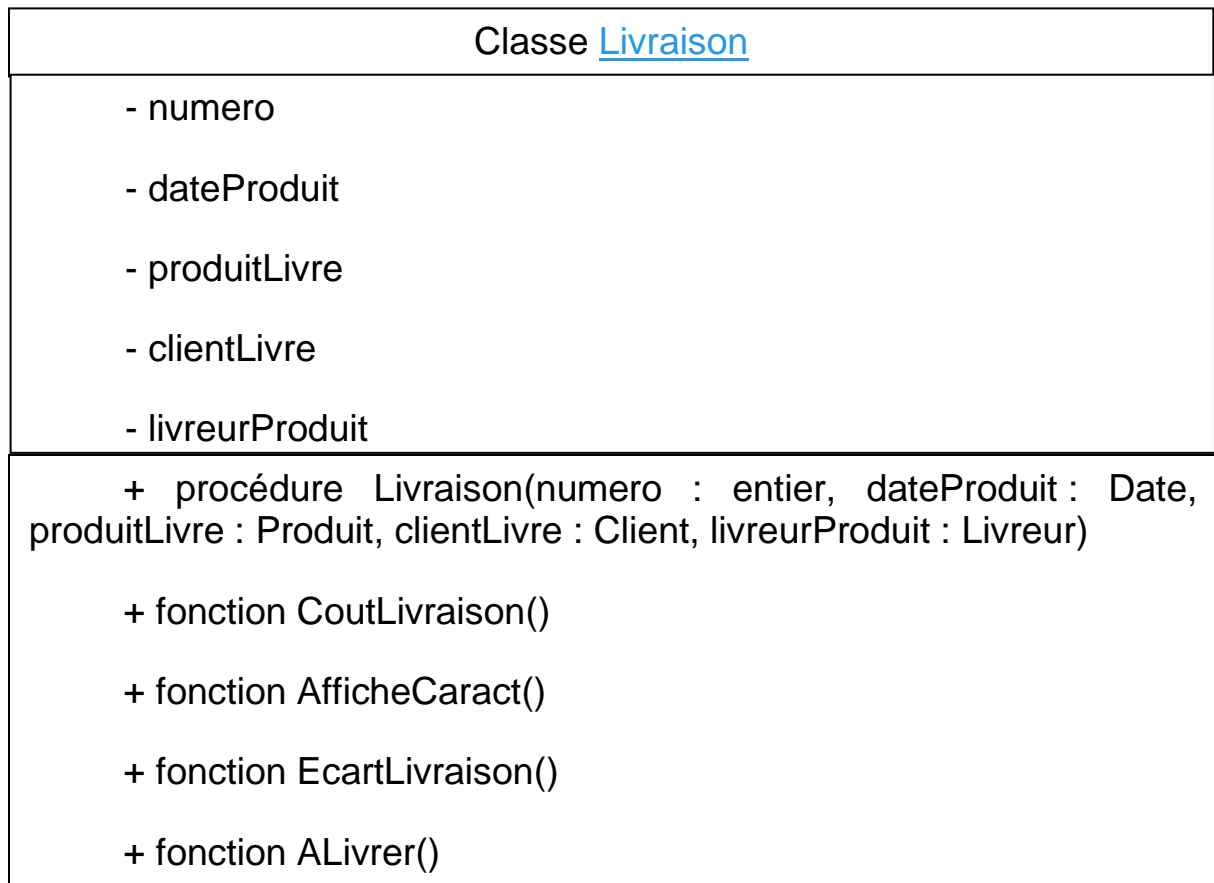
Nous allons abordés la classe [Date](#) particulièrement car c'est la première fois que nous mettons en place une classe comme celle-ci, composé de variables de type DateTime, un type que nous avons évoqué plus tôt.

Ce type permet d'initialiser des variables avec des notions de temp particulières, cela peut passer par les heures, les jours, les mois, les années ou bien tout en même temp. On peut même les afficher de manière différentes, toute en l'heure permettant de se mettre à jour tout seul en se basant sur la date et l'heure de la machine.

Nous avons dû initialiser une fonction date getDatesJour afin de récupérer la date exact au moment où l'on appelle la fonction.

```
0 références
static public Date GetDatesJour()
{
    DateTime djour = DateTime.Today;
    djour.ToString("d");
    return djour;
}
```

## DIAGRAMME UML DE LA CLASSE LIVRAISON



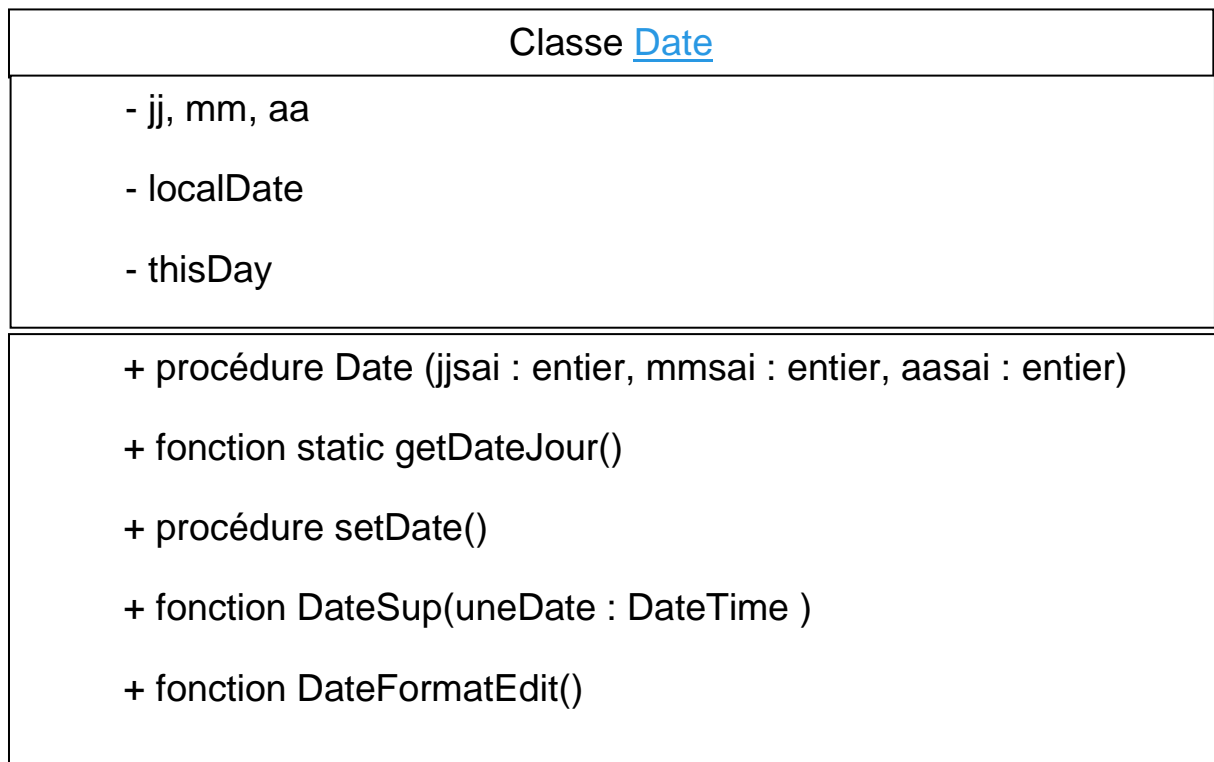
## DIAGRAMME UML DE LA CLASSE CATEGORIE

Classe <a href="#">Categorie</a>
<ul style="list-style-type: none"><li>- nomCateg</li><li>- MajorLivraison</li></ul>
<ul style="list-style-type: none"><li>+ procédure Categorie (nomCateg : chaîne, MajorLivraison : réel)</li><li>+ fonction getNomCateg()</li><li>+ fonction getMajorLivraison()</li><li>+ procédure setNomCateg(nouveauNomCateg : chaîne )</li><li>+ procédure setMajorLivraison(nouveauMajorLivraison : réel)</li></ul>

## DIAGRAMME UML DE LA CLASSE CLIENT

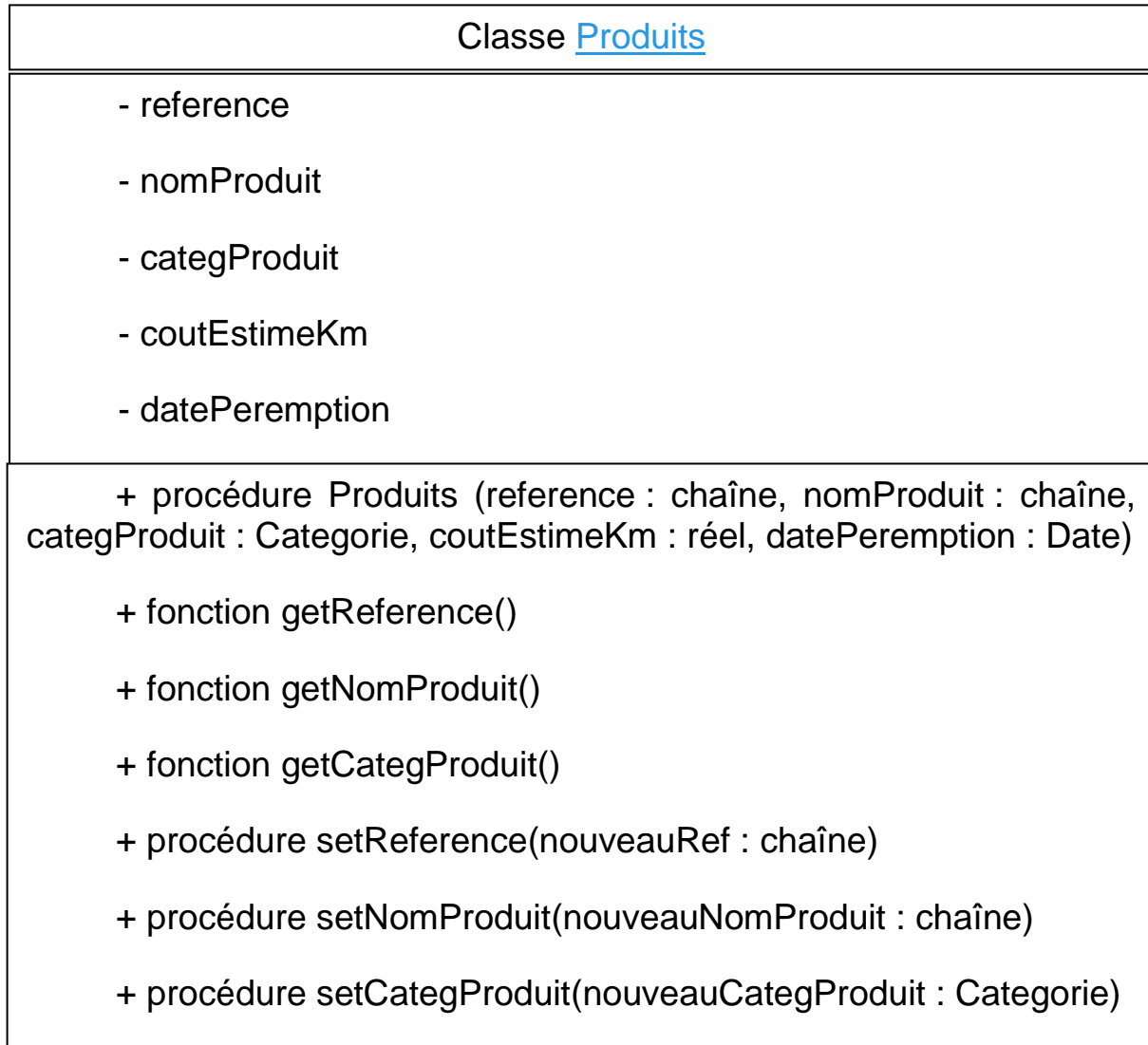
Classe <a href="#">Client</a>
<ul style="list-style-type: none"><li>- nom</li><li>- distance</li></ul>
<ul style="list-style-type: none"><li>+ procédure Client (nom : chaîne, distance : entier)</li><li>+ fonction getNom ()</li><li>+ fonction getDistance()</li><li>+ procédure setNom (nouveauNom : chaîne )</li><li>+ procédure setDistance(nouveauDistance : entier)</li></ul>

## DIAGRAMME UML DE LA CLASSE DATE





## DIAGRAMME UML DE LA CLASSE PRODUITS



## DIAGRAMME UML DE LA CLASSE LIVREUR

Classe <a href="#">Livreur</a>
<ul style="list-style-type: none"><li>- nom</li><li>- coutKm</li></ul>
<ul style="list-style-type: none"><li>+ procédure Livreur (nom : chaîne, coutKm : réel)</li><li>+ fonction getNom ()</li><li>+ fonction getCoutKm()</li><li>+ procédure setNom (nouveauNom : chaîne )</li><li>+ procédure setCoutKm(nouveauCoutKm : réel)</li></ul>

## CONCLUSION

Ces travaux nous ont permis de découvrir les relations entre les classes, comment initialiser dans un constructeur des variables qui proviennent de classe différentes. Cette technique permet de nombreuses simplifications de programmes que nous avons créés auparavant et fait sûrement office d'éléments clés pour la programmation orientée objet.

De plus, nous avons fait une première approche avec le type `DateTime`, qui permet d'afficher la date ou l'heure précise au moment de la requête, ce qui va maintenant nous permettre de l'afficher en temps réel au lieu de la rentrer nous-même dans le code lors de la rédaction du programme.

## ANNEXE

### Classe Livre

```
class Livre
{
    private int numero;
    private string titre;
    private static int compteurLivre;
    private Auteur auteurLivre;

    1 référence
    public Livre (string titre, Auteur auteur)
    {
        this.titre = titre;
        augmentCompteur();
        this.numero = Livre.getCompteur();
        auteurLivre = auteur;
    }

    1 référence
    static public int getCompteur()
    {
        return compteurLivre;
    }

    0 références
    public int getNumero()
    {
        return numero;
    }

    1 référence
    public string getTitre()
    {
        return titre;
    }

    0 références
    private void initCompteur()
    {
        compteurLivre = 0;
    }

    1 référence
    private void augmentCompteur()
    {
        compteurLivre++;
    }
}
```

## Classe Auteur

```
7 références
class Auteur
{
    private string nom;
    private string prenom;
    private MaisonEdition nomMaisonEdition;

    0 références
    public Auteur (string nom, string prenom)
    {
        this.nom = nom;
        this.prenom = prenom;
    }

    1 référence
    public Auteur(string nom, string prenom, MaisonEdition maisonEdition )
    {
        this.nom = nom;
        this.prenom = prenom;
        nomMaisonEdition = maisonEdition;
    }

    1 référence
    public string getNom()
    {
        return nom;
    }

    1 référence
    public string getPrenom()
    {
        return prenom;
    }

    0 références
    public void setNom(string nouveauNom)
    {
        this.nom = nouveauNom;
    }

    0 références
    public void setPrenom(string nouveauPrenom)
    {
        this.prenom = nouveauPrenom;
    }
}
```

```
0 références
public MaisonEdition getNomMaisonEdition()
{
    return nomMaisonEdition;
}

0 références
public void setNomMaisonEdition(MaisonEdition nouveauNomMaisonEdition)
{
    nomMaisonEdition = nouveauNomMaisonEdition;
}
```

## Classe MaisonEdition

```
6 références
class MaisonEdition
{
    private string nom;
    private int cotation;

    1 référence
    public MaisonEdition (string nom, int cotation)
    {
        this.nom = nom;
        this.cotation = cotation;
    }

    0 références
    public string getNom()
    {
        return nom;
    }

    0 références
    public int getCotation()
    {
        return cotation;
    }

    0 références
    public void setNom(string nouveauNom)
    {
        this.nom = nouveauNom;
    }

    0 références
    public void setCotation(string nouveauCotation)
    {
        this.cotation = nouveauCotation;
    }
}
```

## Classe Livraisons

```
class Livraison
{
    private int numero;
    private Date dateLivraison;
    private Produit produitLivre;
    private Client clientLivre;
    private Livreur livreurProduit;

    0 références
    public Livraison (int numero, Date dateLivraison, Produit produitLivre, Client clientLivre, Livreur livreurProduit)
    {
        this.numero = numero;
        this.dateLivraison = dateLivraison;
        this.produitLivre = produitLivre;
        this.clientLivre = clientLivre;
        this.livreurProduit = livreurProduit;
    }

    1 référence
    public double CoutLivraison()
    {
        double km, major, cout;

        major = produitLivre.GetNomProduit().GetMajorLivraison();
        km = clientLivre.GetDistances() * 2;
        cout = livreurProduit.GetCoutKm();

        return major * km * cout;
    }

    1 référence
    public void AfficheCaract()
    {
        Console.WriteLine("Numéro de livraison : ", numero);
        Console.WriteLine("Date de livraison : ", dateLivraison.DateFormatEdit());
        Console.WriteLine("Client : ", clientLivre.GetNom());
        Console.WriteLine("Produit livré : ", produitLivre.GetNomProduit());
        Console.WriteLine("Catégorie de produit : ", produitLivre.GetCateg().GetNomCateg());
        Console.WriteLine("Date de péremption : ", produitLivre.GetDatePeremption().DateFormatEdit());
        Console.WriteLine("Nom livreur : ", livreurProduit.GetNom());
        Console.WriteLine("Coût de la livraison : ", this.CoutLivraison(), "€");
    }
}
```

```
0 références
public double EcartLivraison()
{
    double km, major, coutE, coutR;

    major = produitLivre.GetCateg().GetMajorLivraison();
    km = clientLivre.GetDistance() * 2;
    coutR = livreurProduit.GetCoutKm();
    coutE = produitLivre.GetCoutEstimeKm();

    return km * (coutR - coutE) * major;
}

0 références
public bool ALivrer()
{
    Date djour;
    djour = Date.GetDateJour();

    if ((dateLivraison.DateSup(djour) == true) && (dateLivraison.GetAnnee() == djour.GetAnnee()))
    {
        return true;
    }
    else
    {
        return false;
    }
}
```

## Classe Produits

```
1 reference
class Produits
{
    private string reference;
    private string nomProduit;
    private Categorie categProduit;
    private double coutEstimeKm;
    private Date datePeremption;

    0 références
    public Produits(string referenceSai, string nomProduitSai, double coutEstimekmSai, Date datePreemptionSai)
    {
        this.reference = referenceSai;
        this.nomProduit = nomProduitSai;
        this.coutEstimeKm = coutEstimekmSai;
        this.datePeremption = datePreemptionSai;
    }

    0 références
    public void SetReference(string reference)
    {
        this.reference = reference;
    }

    0 références
    public string GetReference()
    {
        return reference;
    }

    0 références
    public void SetNomProduit(string nomproduit)
    {
        this.nomProduit = nomproduit;
    }

    0 références
    public string GetNomProduit()
    {
        return nomProduit;
    }
}
```



## Les classes composé d'objets

```
0 références
public void SetCateg(Categorie categproduit)
{
    categProduit = categproduit;
}

0 références
public Categorie GetCategs()
{
    return categProduit;
}

0 références
public void SetCoutEstimeKm(double coutEstimeKm)
{
    this.coutEstimeKm = coutEstimeKm;
}

0 références
public double GetCoutEstimeKm()
{
    return coutEstimeKm;
}

0 références
public void SetDatePeremption(Date Dateperemption)
{
    DatePeremption = Dateperemption;
}

0 références
public Date GetDatePeremption()
{
    return DatePeremption;
}
}
```

## Classe Date

```
class Date
{
    private int jj, mm, aa;
    private string localdate;
    private static DateTime thisday = DateTime.Today;

    public Date(int jjsai, int mmsai, int aasai)
    {
        this.jj = jjsai;
        this.mm = mmsai;
        this.aa = aasai;
        SetDate();
    }

    public static DateTime GetDateJour()
    {
        return thisday;
    }

    public void SetDate()
    {
        localdate = $"{jj} / {mm} / {aa}";
    }

    public bool DateSup(DateTime unedate)
    {
        if (unedate < thisday)
        {
            return true;
        }
        else return false;
    }

    public string DateFormatEdit()
    {
        return localdate;
    }

    public int GetAnnee()
    {
        return aa;
    }
}
```

```
public int GetMois()
{
    return mm;
}

public int GetJour()
{
    return jj;
}
```

## Classe Client

```
class Client
{
    private string nom;
    private int distance;

    public Client(string nomSai, int distanceSai)
    {
        this.nom = nomSai;
        this.distance = distanceSai;
    }

    public double GetDistances()
    {
        return distance;
    }

    public string GetNoms()
    {
        return nom;
    }

    public void SetNom(string nom)
    {
        this.nom = nom;
    }

    public void SetDistance(int distance)
    {
        this.distance = distance;
    }
}
```

## Classe Livreur

```
class Livreur
{
    private string nom;
    private double coutkm;

    public Livreur(string nomSai, double coutkmSai)
    {
        this.nom = nomSai;
        this.coutkm = coutkmSai;
    }

    public double GetCoutKm()
    {
        return coutkm;
    }

    public string GetNom()
    {
        return nom;
    }

    public void SetCoutKm(double coutkm)
    {
        this.coutkm = coutkm;
    }

    public void SetNoms(string nom)
    {
        this.nom = nom;
    }
}
```

## Classe Cat  gorie

```
class Categorie
{
    private string nomCateg;
    private double MajorLivraison;

    public Categorie(string nomCategSai, double majorLivraisonSai)
    {
        this.nomCateg = nomCategSai;
        this.MajorLivraison = majorLivraisonSai;
    }

    public string GetNomCateg()
    {
        return nomCateg;
    }

    public double GetMajorLivraison()
    {
        return MajorLivraison;
    }

    public void SetNomCateg(string NomCateg)
    {
        this.nomCateg = NomCateg;
    }

    public void SetMajorLivraison(double MajorLivraison)
    {
        this.MajorLivraison = MajorLivraison;
    }
}
```