



SIVY
Société d'informatique de Vichy

- Gestion interne du musée d'art des Célestins

PRESENTATION

Le musée des Célestins est un musée situé à Vichy. Le musée possède une des plus grandes collections d'œuvres impressionnistes et postimpressionnistes au monde.

On peut y voir des chefs-d'œuvre de la peinture comme « Le Déjeuner sur l'herbe » et « l'Olympia » d'Édouard Manet, « la Danseuse âgée de quatorze ans » de Degas,

Plus de 5 000 peintures composent la collection.



- **Pour chaque ŒUVRE, les informations relatives à l'ARTISTE sont mémorisées.**

Exemple :



Le Déjeuner sur l'herbe est un tableau d'Édouard Manet datant de 1863.

« Édouard Manet (né à Paris le 23 janvier 1832 - mort à Paris le 30 avril 1883) est un peintre français majeur de la fin du XIXe siècle... »



- **Les ŒUVRES sont exposées dans des SALLES.**

Chaque salle est « thématique » puisque elle héberge un ensemble d'œuvres ayant un ou plusieurs points communs. La composition du musée en salles est « dynamique » dans la mesure où elles vont évoluer au gré des expositions organisées par la direction du musée. Par exemple, au jour d'aujourd'hui, le musée est organisé autour des salles suivantes : Ecole « française », Ecole « Néerlandaise », Exposition MANET, Exposition DEGAS, etc...

Exemple : Salle « Exposition MANET »



Manet, Edouard
Berthe Morisot au
bouquet de violettes



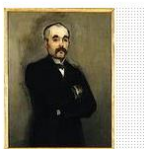
Manet, Edouard
Portrait de Madame
Edouard Manet sur un
canapé bleu



Manet, Edouard
Portrait d'Irma
Brunner



Manet, Edouard
Le tub



Manet, Edouard
Georges Clemenceau



Manet, Edouard
Sur la plage

La Direction du musée souhaite pouvoir connaître à tout moment :

- a) La COMPOSITION du musée :
 - La liste des salles avec les œuvres qu'y figurent.
 - Sa collection d'œuvres, toutes salles confondues.
 - L'ensemble des artistes références, toutes œuvres (et salles) confondues.
- b) Pour chaque salle, l'ECART, par salle, entre la VALEUR ESTIMEE de l'ensemble des œuvres et le montant de l'ASSURANCE contractée par le musée.

Par ailleurs, le musée expose deux types d'œuvres :

- Des œuvres ACHETEES par le musée, dont on doit connaître, entre autres, leur prix et leur date d'achat.
- Des œuvres PRETEES par d'autres musées ou des collectionneurs, dont on souhaite connaître le nom du prêteur ainsi que les dates de début et de fin de prêt, ce qui permettra de préciser le nombre de jours pendant lesquels l'œuvre restera exposée dans le musée.

On souhaite intégrer la « maintenance » (expertise et vérification de l'état) des œuvres ACHETEES.

Pour chaque œuvre achetée (= propriété du musée), on attribue un « état » (=bon, moyen, mauvais). L'état de l'œuvre permet de spécifier le nombre de jours maximum toléré, s'écoulant entre deux expertises. Par exemple, une œuvre dont l'état est « mauvais » doit être vérifiée tous les 30 jours.

Lors d'une vérification effectuée par un expert interne ou externe au musée, on détermine la ou les œuvres devant faire l'objet d'une restauration. Dans ce cas, les œuvres concernées sont retirées des salles dans laquelle elles sont exposées et placées dans un local « ad-hoc ». La gestion des experts et des restaurations ne concerne pas la version actuelle du projet.

En outre, on devra attribuer à chaque œuvre achetée, la date de la DERNIERE expertise et un INDICATEUR permettant de savoir si l'œuvre est à expertiser ou non. Par exemple, soit l'œuvre « Tournesols dans un vase » de Vincent Van Gogh :

- Elle a été achetée le 15 septembre 2021. Sa dernière expertise a été effectuée (ici, par son ancien propriétaire) le 15 juillet 2021. Son état est considéré « mauvais » (=une expertise est donc nécessaire tous les 30 jours).
 - L'œuvre est à expertiser si l'écart en nombre de jours, entre la date actuelle est la date de sa dernière expertise est supérieur ou égal au nombre de jours séparant deux expertises (=fonction de l'état de l'œuvre).
- L'écart pour cette œuvre est supérieur à 30 jours (15/09 – 15/07). L'œuvre est donc à expertiser !

Pour chaque musée, il faudra prévoir un service qui fournit, par SALLE, l'ensemble des ŒUVRES devant faire l'objet d'une expertise, en fonction de la date à laquelle ce service est utilisé.

REMARQUE : pour le PLAN DE TESTS (voir plus loin) on va considérer que l'ECART se calcule entre la DATE d'ACHAT (et non pas entre la DATE SYSTEME) et la DATE DE LA DERNIERE EXPERTISE.

L'APPLICATION A DEVELOPPER

La Direction du Musée des Célestins s'est adressée à l'ESN « SIVY » pour le développement du logiciel correspondant. Un contrat dit « au FORFAIT » (=durée de réalisation et coût associé estimés et négociés avant le démarrage du projet) a été signé entre les deux parties :

- a) En fonction des besoins exprimés, sous la forme d'un cahier de charges (non fourni dans ce projet), le responsable du Pôle développement de SIVY, a réalisé une estimation de la charge de travail nécessaire.
- b) Un contrat a alors été établi entre les deux parties dans lequel figurent, entre autres :
 - Le détail de la prestation et le découpage en lots techniques.
 - Pour chaque lot : la charge de travail estimée, les délais correspondants, le montant facturé, et enfin, les modalités de paiement et de livraison.
- c) Le développement s'effectue dans les locaux de SIVY.
- d) Le chef de projet est en charge du suivi de projet et de l'interface entre le client et son équipe de développement.

Le logiciel à développer constituera une application « pilote » qui, une fois validée, pourra s'étendre à l'ensemble des musées d'art relevant du Ministère de la Culture. Ceci explique la présence de la classe métier « MUSEE ». La plateforme retenue est le Framework .NET et le langage C#.

MISSION 1

Il s'agit de développer une application CONSOLE.

- Un DIAGRAMME de CLASSES METIER est fourni (cf. annexe 1).
- Une implémentation partielle des CLASSES METIER et TECHNIQUES (=déclaration des PREDICATS et des méthodes de COMPARAISON à appliquer sur les classes métier) doit être complétée (cf. annexe 2 et le code fourni).

IMPORTANT : ces classes devront être INDEPENDANTES de l'IHM (console, Windows Forms, web, mobile, etc...) – cf. MISSION 2.

- Un PLAN de TEST a été établi par votre chef de projet. Il devra être utilisé afin de vérifier la conformité du code développé par rapport aux résultats attendus (cf. annexe 3).
- La dernière partie du plan de test devra être écrite : il s'agit de tester la méthode `public Dictionary<Salle, List<Oeuvre_Achetee>> getLesExpertisesAFaire()` de la classe MUSEE, puis d'exploiter le retour de cette méthode (cf. DICTIONNAIRE).

MISSION 2

Dans l'objectif d'automatiser les TEST UNITAIRES, vous devez créer un PROJET DE TEST UNITAIRES dans le but de tester la méthode `public Dictionary<Salle, List<Oeuvre_Achetee>> getLesExpertisesAFAire()` de la classe MUSEE.

Vous écrirez les méthodes de test qui vous sembleraient opportunes afin de tester la méthode précédente.

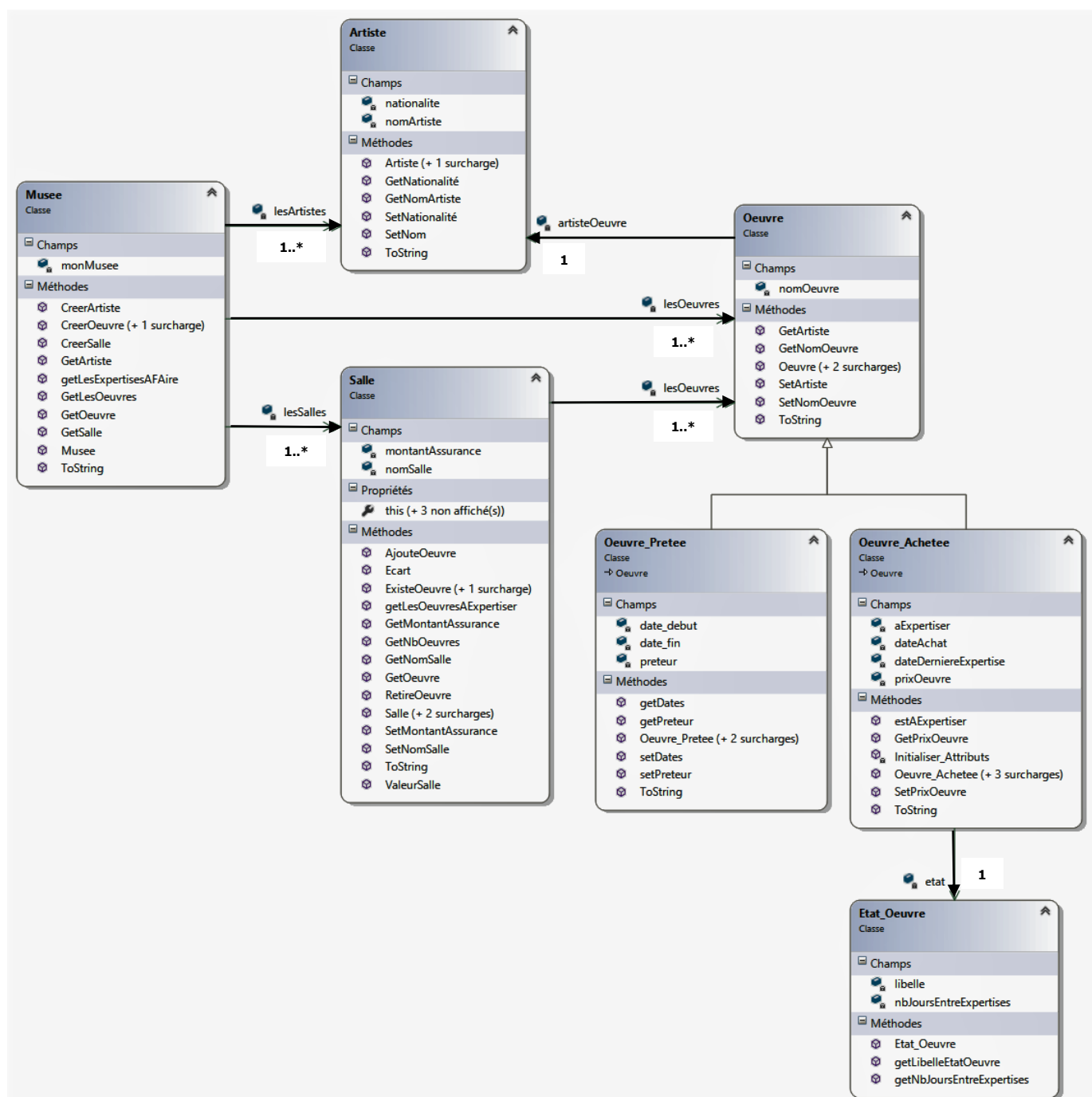
MISSION 3

Il s'agit de vérifier que l'ensemble de classes complétées sont indépendantes de l'interface du futur logiciel.

Il vous est demandé de concevoir, puis de développer une APPLICATION GRAPHIQUE Windows Forms :

- Vous devrez intégrer à l'application l'ensemble des classes METIERS et/ou TECHNIQUES de la mission 1, SANS les MODIFIER.
- Vous pouvez utiliser les données mises en œuvre dans le PLAN DE TEST précédemment fourni.
- Vous avez le CHOIX quant à la nature et aux services offerts par la nouvelle application, à condition qu'elle utilise de manière significative les services offerts par les classes précédentes et que la fonctionnalité présentée s'inscrive dans un contexte d'IHM graphique.

ANNEXE 1 : diagramme des classes métier



ANNEXE 2 : classes métier et technique

```
public class Artiste
{
    // Attributs
    private string nomArtiste;
    private string nationalite;

    // Constructeur (+ surcharge)
    3 références
    public Artiste(string nom, string nat)...
    3 références
    public Artiste(Artiste a)...

    // Accesseurs
    9 références
    public string GetNomArtiste()...
    1 référence
    public string GetNationalité()...
    0 références
    public void SetNom(string nom)...
    0 références
    public void SetNationalité(string nat)...

    // Retourne une chaine avec nom et nationalité de l'artiste
    5 références
    public override string ToString()...
}
```

```
public class Etat_Oeuvre
{
    // Attributs
    private string libelle;
    private int nbJoursEntreExpertises;

    // Constructeur
    3 références
    public Etat_Oeuvre(string l, int nb)...

    // Accesseurs
    1 référence
    public string getLibelleEtatOeuvre()...
    1 référence
    public int getNbJoursEntreExpertises()...
}
```

```
public class Oeuvre
{
    // Attributs communs
    private string nomOeuvre;
    private Artiste artisteOeuvre;

    // Constructeur (+ surcharges)
    3 références
    public Oeuvre(string nom)...
    2 références
    public Oeuvre(string nom, Artiste a)...
    4 références
    public Oeuvre(Oeuvre o)...

    // Accesseurs
    10 références
    public string GetNomOeuvre()...
    0 références
    public void SetNomOeuvre(string nom)...

    // Accesseurs sur l'attribut 'artisteOeuvre'
    5 références
    public Artiste GetArtiste()...
    2 références
    public void SetArtiste(Artiste a)...

    // Retourne une chaine avec les caractéristiques de l'oeuvre avec son artiste.
    // Puisque cette méthode doit être REDÉFINIE dans les classes DERIVEES, et
    // qu'elle est HÉRITÉE de la classe OBJECT, il faut la déclarer en tant que
    // "new", ce qui indique qu'elle ne la redéfinit pas mais la "masque".
    14 références
    public virtual new string ToString()...
}
```

```
public class Oeuvre_Pretée : Oeuvre
{
    // Attribut spécifique
    private string preteur;
    private DateTime date_debut;
    private DateTime date_fin;

    // Constructeur (+surcharges)
    // Pour la dernière surcharge :
    // * Les dates seront initialisées à la date courante
    // * Le prêteur sera initialisé à "inconnu".
    3 références
    public Oeuvre_Pretée(string nom, string preteur, DateTime debut, DateTime fin)...
    0 références
    public Oeuvre_Pretée(string nom, string preteur, DateTime debut, DateTime fin, Artiste a)...
    0 références
    public Oeuvre_Pretée(Oeuvre o)...

    // Accesseurs. Remarque : getDates retourne un TABLEAU avec les deux dates (début et fin)
    // et setDates reçoit en paramètre un tableau analogue.
    1 référence
    public string getPreteur()...
    2 références
    public DateTime[] getDates()...
    0 références
    public void setPreteur(string p)...
    0 références
    public void setDates(DateTime[] dates)...

    // Méthode REDÉFINIE. Formatage une chaine avec les informations de l'oeuvre.
    14 références
    public override string ToString()...
}
```

```

public class Oeuvre_Achetee : Oeuvre
{
    // Attributs spécifiques
    private float prixOeuvre;
    private DateTime dateAchat;
    private Etat_Oeuvre etat;
    private DateTime dateDerniereExpertise;
    private bool aExpertiser;

    // Constructeur (+surcharges)
    4 références
    public Oeuvre_Achetee(string nom, float prix, DateTime achat, DateTime expertise, Etat_Oeuvre etat)[...]
    0 références
    public Oeuvre_Achetee(string nom, float prix, Etat_Oeuvre etat) [...]
    6 références
    public Oeuvre_Achetee(string nom, float prix, DateTime achat, DateTime expertise, Etat_Oeuvre etat, Artiste a)[...]
    0 références
    public Oeuvre_Achetee(Oeuvre o)[...]

    // Initialise les attributs spécifiques. Appellée par les constructeurs.
    4 références
    private void Initialiser_Attributs(float prix, DateTime achat, DateTime expertise, Etat_Oeuvre e)[...]

    // Accesseurs
    4 références
    public float GetPrixOeuvre()[...]
    0 références
    public void SetPrixOeuvre(float prix)[...]

    // Méthode permettant de savoir si une oeuvre est à expertiser.
    3 références
    public bool estAExpertiser()[...]

    // Méthode REDEFINIE. Formatage une chaine avec les informations de l'oeuvre.
    14 références
    public override string ToString()[...]
}

```

```

public class Salle
{
    // Attributs
    private string nomSalle;
    private float montantAssurance;
    private List<Oeuvre> lesOeuvres; // Objets POLYMORPHES

    // Constructeur (+ surcharges)
    2 références
    public Salle(string nomSalle, float montant)[...]
    0 références
    public Salle(Salle s)[...]
    0 références
    public Salle(string nomSalle, float montant, List<Oeuvre> l)[...]

    // Accesseurs sur le nom de la salle et le montant de l'assurance.
    6 références
    public string GetNomSalle()[...]
    0 références
    public float GetMontantAssurance()[...]
    0 références
    public void SetNomSalle(string nom)[...]
    0 références
    public void SetMontantAssurance(float mtt)[...]

    // Retourne l'oeuvre dont le nom est passé en paramètre ou "null" sinon trouvée.
    0 références
    public Oeuvre GetOeuvre(string nom)[...]

    // Retourne vrai si l'Oeuvre dont le nom est passé en paramètre existe dans la salle,
    // faux sinon (+ surcharge).
    2 références
    public bool ExisteOeuvre(string nom)[...]
    2 références
    public bool ExisteOeuvre(Oeuvre uneOeuvre)[...]

    INDEXEURS

    // Ajoute une Oeuvre dans la salle
    // Retourne vrai si l'ajout a eu lieu, faux si l'oeuvre existe déjà
    2 références
    public bool AjouteOeuvre(Oeuvre uneOeuvre)[...]

    // Enlève une Oeuvre dans la salle
    // Retourne vrai si le retrait a eu lieu, faux si l'oeuvre n'existe pas
    0 références
    public bool RetireOeuvre(Oeuvre uneOeuvre)[...]

    // Retourne la valeur totale des oeuvres stockées dans la salle
    3 références
    public double ValeurSalle()[...]

    // Retourne l'écart entre le montant de la valeur déclarée
    // à l'assurance et la valeur des Oeuvres
    // Lorsque la valeur de la salle est égale à ZERO
    // (=que des oeuvres prêtées) alors on retourne ZERO
    1 référence
    public double Ecart()[...]

    // Retourne l'ensemble des oeuvres devant faire l'objet d'une expertise
    1 référence
    public List<Oeuvre_Achetee> getLesOeuvresAExpertiser()[...]

    // Retourne une chaine avec les caractéristiques de la salle
    // (nom et montant de l'assurance) et des œuvres avec artiste.
    5 références
    public override string ToString()[...]
}

```

```

public class Musee
{
    // Attributs
    private string monMusee;           // Nom du musée
    private List<Artiste> lesArtistes;  // Liste des artistes
    private List<Salle> lesSalles;      // Liste des salles
    private List<Oeuvre> lesOeuvres;    // Liste des oeuvres;

    // Constructeur : création d'un musée
    1 référence
    public Musee(string nom) {...}

    // Création d'une SALLE
    2 références
    public string CreerSalle(Salle s) {...}

    // Création d'un ARTISTE
    3 références
    public string CreerArtiste(Artiste a) {...}

    // Création d'une OEUVRE AVEC UN ARTISTE ET UNE SALLE
    // (+ surcharge : OEUVRE AVEC L'ARTISTE DÉJÀ DÉFINI)
    7 références
    public string CreerOeuvre(Oeuvre o, Artiste a, Salle s) {...}
    6 références
    public string CreerOeuvre(Oeuvre o, Salle s) {...}

    // Accesseurs
    0 références
    public List<Oeuvre> GetLesOeuvres() {...}
    0 références
    public List<Salle> GetLesSalles() {...}
    0 références
    public List<Artiste> GetLesArtistes() {...}
    0 références
    public Artiste GetArtiste(int i) {...}
    0 références
    public Oeuvre GetOeuvre(int i) {...}
    13 références
    public Salle GetSalle(int i) {...}

    // RETOURNE UNE COLLECTION DICTIONNAIRE AVEC LA LISTE DES OEUVRES
    // devant faire l'objet d'une expertise. Chaque élément est de la forme :
    // * Clé : objet SALLE
    // * Valeur associée : collection d'oeuvres à expertiser.
    1 référence
    public Dictionary<Salle, List<Oeuvre_Achetee>> getLesExpertisesAFAire() {...}

    // Formatte une chaîne avec les données du MUSEE.
    5 références
    public override string ToString() {...}
}

```

REMARQUES

a) La liste des **CONTROLES de cohérence des données est PARTIELLE**. A vous d'identifier et implémenter des contrôles supplémentaires permettant de garantir la cohérence des données.

b) Classe dérivée **ŒUVRE_ACHETEE** :

- Surcharge du constructeur `public Oeuvre_Achetee(Oeuvre o)`

Si le paramètre est un objet de classe **ŒUVRE_ACHETEE** (cf. aide technique ci-après), on peut donc affecter son prix à l'attribut correspondant `this.prixOeuvre`. Sinon, celui-ci sera initialisé à zéro. On vous fournit les renseignements suivants :

is (référence C#)

Visual Studio 2010 | [Autres versions](#) ▼

Vérifie si un objet est compatible avec un type donné. Par exemple, le code suivant permet de déterminer si un objet est une instance du type **MyObject** ou un type dérivé de **MyObject** :

```

if (obj is MyObject)
{
}

```

Une expression **is** prend la valeur **true** si l'expression fournie n'est pas null, et que l'objet fourni peut être casté en type fourni sans entraîner la levée d'une exception.

- Pour l'autre classe dérivée **ŒUVRE_PRETEE**, le principe sera le même. Si le paramètre n'est pas un objet de la classe **Oeuvre_Prete**, on initialisera le prêteur à « inconnu » et les dates à la date système actuelle.

c) Classe SALLE

- Il s'agit de proposer, en plus, des services de recherche sous la forme d'INDEXEURS en LECTURE (=accès à(aux) œuvre(s) de la salle) :
 - sur le rang de l'œuvre dans la liste ;
 - sur le prix (=toutes les œuvres dont le prix est \leq à celui cherché) ;
 - sur l'artiste (=objet ARTISTE), puis son nom (=toutes les œuvres de l'artiste cherché).
- Une ŒUVRE qui est déjà présente dans la salle ne peut pas lui être ajoutée une deuxième fois. Inversement, on ne peut retirer une ŒUVRE que si elle figure dans la salle.
- La méthode `double ValeurSalle()` retourne la valeur totale des tableaux exposés dans la salle. Cela ne concerne, naturellement, que les œuvres achetées par le musée, dont on connaît leur prix.
- La méthode `double Ecart()` retourne l'écart entre le montant de la valeur déclarée à l'assurance et la valeur des Œuvres. Lorsque la valeur de la salle est égale à ZERO (=elle ne contient que des œuvres prêtées) alors on retourne ZERO.

d) Classe MUSEE

- Lorsque l'on passe un nom pas renseigné (=chaîne vide) au constructeur `Musee(string nom)`, l'attribut correspondant prendra la valeur « inconnu ».
- Lorsque l'on ajoute une SALLE, un ARTISTE ou une OEUVRE au musée, il ne faudra pas accepter des doublons.
- La surcharge `string CreerOeuvre(Oeuvre o, Salle s)` concerne une ŒUVRE pour laquelle un ARTISTE a déjà été défini ; c'est pourquoi celui-ci n'est pas transmis en tant que 3^{ème} paramètre.

e) Une classe « TECHNIQUE » est ajoutée au projet. Il s'agit de proposer à l'utilisateur des éléments permettant d'effectuer des traitements prédéfinis dans le .NET avec les collections : « Find », « FindAll », « RemoveAll »... « Sort », etc...

```
// UTILISATION DES PREDICATS ET DE LA METHODE DE COMPARAISON SUR LES CLASSES METIER
5 références
public class Predicats
{
    // Donnée utilisées par le PREDICAT
    public static string nomArtiste = "";

    // Méthode PREDICAT (pour "Find()", "FindAll()"...)
    // Cette fonction sera appliquée, à tour de rôle, à chaque élément d'une collection d'OEUVRES pour une SALLE.
    1 référence
    public static bool RechercheOeuvresArtiste(Oeuvre o)...

    // Méthodes de COMPARAISON (pour "Sort()") entre deux objets OEUVRE ACHETEE.
    // Cette méthode doit retourner, suivant le critère de comparaison :
    //      0 si = égalité
    //      1 si o1 > o2
    //     -1 si o1 < o2
    1 référence
    public static int ComparerOeuvresParNom(Oeuvre o1, Oeuvre o2)...
    1 référence
    public static int ComparerOeuvresParPrix(Oeuvre o1, Oeuvre o2)...
}
```

On vous fournit les renseignements techniques suivants :

List<T>.FindAll, méthode

.NET Framework 4 | [Autres versions ▼](#)

Récupère tous les éléments qui correspondent aux conditions définies par le prédicat spécifié.

Espace de noms : `System.Collections.Generic`
Assembly : `mscorlib` (dans `mscorlib.dll`)

Syntaxe

```
C# C++ F# VB
public List<T> FindAll(
    Predicate<T> match
)
```

Paramètres

match

Type : `System.Predicate<T>`

Délégué `Predicate<T>` qui définit les conditions des éléments à rechercher.

Valeur de retour

Type : `System.Collections.Generic.List<T>`

`List<T>` contenant tous les éléments qui correspondent aux conditions définies par le prédicat spécifié, si une correspondance est trouvée ; sinon, un `List<T>` vide.

List<T>.Sort, méthode (Comparison<T>)

.NET Framework 4 | [Autres versions](#) ▼

Trie les éléments dans l'intégralité de [List<T>](#) à l'aide du [System.Comparison<T>](#) spécifié.

Espace de noms : [System.Collections.Generic](#)
Assembly : mscorlib (dans mscorlib.dll)

Syntaxe

C# C++ F# VB

```
public void Sort(  
    Comparison<T> comparison  
)
```

Paramètres

comparison
Type : [System.Comparison<T>](#)
[System.Comparison<T>](#) à utiliser lors de la comparaison d'éléments.

Paramètres de type

in T
Type des objets à comparer.

Ce paramètre de type est contravariant. Autrement dit, vous pouvez utiliser le type que vous avez spécifié ou tout type moins dérivé. Pour plus d'informations sur la covariance et la contravariance, consultez [Covariance et contravariance dans les génériques](#).

Paramètres

x
Type : T
Premier objet à comparer.

y
Type : T
Second objet à comparer.

Valeur de retour

Type : [System.Int32](#)
Entier signé qui indique les valeurs relatives de *x* et *y*, comme indiqué dans le tableau suivant.

Valeur	Signification
Inférieure à 0	<i>x</i> est inférieur à <i>y</i> .
0	<i>x</i> est égal à <i>y</i> .
Supérieure à 0	<i>x</i> est supérieur à <i>y</i> .

Méthode de COMPARAISON entre deux objets :

Comparison<T>, délégué

.NET Framework 4 | [Autres versions](#) ▼

Représente la méthode qui compare deux objets du même type.

Espace de noms : [System](#)
Assembly : mscorlib (dans mscorlib.dll)

Syntaxe

C# C++ F# VB

```
public delegate int Comparison<in T>(  
    T x,  
    T y  
)
```


ANNEXE 3 : plan de test et résultats attendus

```
#region CREER UN MUSEE

string nom = "Musee des Celestins - VICHY";
Musee celestins = new Musee(nom);
Console.WriteLine("*** Création du musee {0} ***\n", nom);

// -- Ajouter des ARTISTES
Artiste monet = new Artiste("Monet", "Francaise");
Artiste manet = new Artiste("Manet", "Francaise");
Artiste vangogh = new Artiste("Van Gogh", "Neerlandaise");
Console.WriteLine(celestins.CreerArtiste(monet));
Console.WriteLine(celestins.CreerArtiste(manet));
Console.WriteLine(celestins.CreerArtiste(vangogh));

// -- Ajouter des SALLES
Salle francaise = new Salle("Francaise", 200000);
Salle neerlandaise = new Salle("Neerlandaise", 200000);
Console.WriteLine(celestins.CreerSalle(francaise));
Console.WriteLine(celestins.CreerSalle(neerlandaise));

// -- Ajouter des ETATS des OEUVRES
Etat_Oeuvre bon = new Etat_Oeuvre("Bon", 365);
Etat_Oeuvre moyen = new Etat_Oeuvre("Moyen", 180);
Etat_Oeuvre mauvais = new Etat_Oeuvre("Mauvais", 30);
```

Création du musée :

- Ajout d'artistes ;
- Ajout de salles ;
- Ajout des œuvres ;
- Affectation des œuvres aux artistes et aux salles.

```
// -- Ajouter des OEUVRES

// MONET
Oeuvre o1 = new Oeuvre_Achetee("Le Déjeuner sur l'herbe", 500000, new DateTime(2021,09,15), new DateTime(2021, 07, 15), moyen, monet);
Oeuvre o2 = new Oeuvre_Achetee("Au bord de l'eau", 350000, new DateTime(2021, 09, 15), new DateTime(2021, 05, 15), bon);
Oeuvre o3 = new Oeuvre_Achetee("La Plage à Honfleur", 90000, new DateTime(2021, 09, 15), new DateTime(2021, 07, 1), mauvais, monet); // A expertiser
Oeuvre o4 = new Oeuvre_Achetee("Quai du Louvre", 110000, new DateTime(2021, 09, 15), new DateTime(2021, 07, 15), mauvais, monet); // A expertiser

// MANET
Oeuvre o5 = new Oeuvre_Pretée("La Partie de croquet", "Musée d'Orsay", new DateTime(2021,05,03), new DateTime(2022,01,15));
Oeuvre o6 = new Oeuvre_Pretée("Le Joueur de fifre", "Musée du Prado", new DateTime(2021, 08, 01), new DateTime(2022, 12, 31));
Oeuvre o7 = new Oeuvre_Achetee("Le petit Lange", 110000, new DateTime(2021, 09, 15), new DateTime(2021, 08, 25), mauvais, manet);
Oeuvre o8 = new Oeuvre_Achetee("L'Exécution de Maximilien", 250000, new DateTime(2021, 09, 15), new DateTime(2021, 08, 1), mauvais, manet); // A expertiser
Oeuvre o9 = new Oeuvre_Achetee("Portrait d'Émile Zola", 95000, new DateTime(2021, 09, 15), new DateTime(2021, 06, 15), moyen, manet);

// VAN GOGH
Oeuvre o10 = new Oeuvre_Achetee("Tournesols dans un vase", 750000, new DateTime(2021, 09, 15), new DateTime(2021, 07, 15), mauvais); // A expertiser
Oeuvre o11 = new Oeuvre_Pretée("Champ de ble avec cypres", "Metropolitan Museum", new DateTime(2020, 10, 24), new DateTime(2021, 10, 24));
Oeuvre o12 = new Oeuvre_Achetee("Autoportrait", 1000000, new DateTime(2021, 09, 15), new DateTime(2021, 08, 30), mauvais);
Oeuvre o13 = new Oeuvre_Achetee("Le Champ de blé aux iris", 95000, new DateTime(2021, 09, 15), new DateTime(2021, 07, 15), moyen);
```

```
// SALLE FRANCAISE - MONET
Console.WriteLine(celestins.CreerOeuvre(o1, celestins.GetSalle(0)));
Console.WriteLine(celestins.CreerOeuvre(o2, monet, celestins.GetSalle(0)));
Console.WriteLine(celestins.CreerOeuvre(o3, celestins.GetSalle(0)));
Console.WriteLine(celestins.CreerOeuvre(o4, celestins.GetSalle(0)));

// SALLE FRANCAISE - MANET
Console.WriteLine(celestins.CreerOeuvre(o5, manet, celestins.GetSalle(0)));
Console.WriteLine(celestins.CreerOeuvre(o6, manet, celestins.GetSalle(0)));
Console.WriteLine(celestins.CreerOeuvre(o7, celestins.GetSalle(0)));
Console.WriteLine(celestins.CreerOeuvre(o8, celestins.GetSalle(0)));
Console.WriteLine(celestins.CreerOeuvre(o9, celestins.GetSalle(0)));

// SALLE NEERLANDAISE - VAN GOGH
Console.WriteLine(celestins.CreerOeuvre(o10, vangogh, celestins.GetSalle(1)));
Console.WriteLine(celestins.CreerOeuvre(o11, vangogh, celestins.GetSalle(1)));
Console.WriteLine(celestins.CreerOeuvre(o12, vangogh, celestins.GetSalle(1)));
Console.WriteLine(celestins.CreerOeuvre(o13, vangogh, celestins.GetSalle(1)));

#endregion
```

Résultat attendu :

```
#region AFFICHER LE MUSEE
Console.WriteLine(celestins.ToString());
#endregion
```

```
*****
Musee des Celestins - VICHY
*****

--- Liste des ARTISTES

[Monet, nationalité Francaise]
[Manet, nationalité Francaise]
[Van Gogh, nationalité Neerlandaise]
```

...

--- Liste des OEUVRES

[Le Déjeuner sur l'herbe=> Monet, Française]
Achetée le 15/09/2021 et expertisée le 15/07/2021
Etat Moyen
Ecart 62
Pour 500000 euros

[Au bord de l'eau=> Monet, Française]
Achetée le 15/09/2021 et expertisée le 15/05/2021
Etat Bon
Ecart 123
Pour 350000 euros

[La Plage à Honfleur=> Monet, Française]
Achetée le 15/09/2021 et expertisée le 01/07/2021
Etat Mauvais
Ecart 76
Pour 90000 euros
---> Doit être expertisée

[Quai du Louvre=> Monet, Française]
Achetée le 15/09/2021 et expertisée le 15/07/2021
Etat Mauvais
Ecart 62
Pour 110000 euros
---> Doit être expertisée

[La Partie de croquet=> Monet, Française]
Prêtée par Musée d'Orsay pour 257 jours

[Le Joueur de fifre=> Monet, Française]
Prêtée par Musée du Prado pour 517 jours

[Le petit Lange=> Manet, Française]
Achetée le 15/09/2021 et expertisée le 25/08/2021
Etat Mauvais
Ecart 21
Pour 110000 euros

[L'Exécution de Maximilien=> Manet, Française]
Achetée le 15/09/2021 et expertisée le 01/08/2021
Etat Mauvais
Ecart 45
Pour 250000 euros
---> Doit être expertisée

[Portrait d'Émile Zola=> Manet, Française]
Achetée le 15/09/2021 et expertisée le 15/06/2021
Etat Moyen
Ecart 92
Pour 95000 euros

[Tournesols dans un vase=> Van Gogh, Néerlandaise]
Achetée le 15/09/2021 et expertisée le 15/07/2021
Etat Mauvais
Ecart 62
Pour 750000 euros
---> Doit être expertisée

[Champ de ble avec cypres=> Van Gogh, Néerlandaise]
Prêtée par Metropolitan Museum pour 365 jours

[Autoportrait=> Van Gogh, Néerlandaise]
Achetée le 15/09/2021 et expertisée le 30/08/2021
Etat Mauvais
Ecart 16
Pour 1000000 euros

[Le Champ de blé aux iris=> Van Gogh, Néerlandaise]
Achetée le 15/09/2021 et expertisée le 15/07/2021
Etat Moyen
Ecart 62
Pour 95000 euros

--- Liste des SALLES

SALLE : Française. Montant d'assurance : 2000000
Il y a 9 oeuvre(s)

[Le Déjeuner sur l'herbe=> Monet, Française]
Achetée le 15/09/2021 et expertisée le 15/07/2021
Etat Moyen
Ecart 62
Pour 500000 euros

[Au bord de l'eau=> Monet, Française]
Achetée le 15/09/2021 et expertisée le 15/05/2021
Etat Bon
Ecart 123
Pour 350000 euros

[La Plage à Honfleur=> Monet, Française]
Achetée le 15/09/2021 et expertisée le 01/07/2021
Etat Mauvais
Ecart 76
Pour 90000 euros
---> Doit être expertisée

[Quai du Louvre=> Monet, Française]
Achetée le 15/09/2021 et expertisée le 15/07/2021
Etat Mauvais
Ecart 62
Pour 110000 euros
---> Doit être expertisée

[La Partie de croquet=> Monet, Française]
Prêtée par Musée d'Orsay pour 257 jours

[Le Joueur de fifre=> Monet, Française]
Prêtée par Musée du Prado pour 517 jours

[Le petit Lange=> Manet, Française]
Achetée le 15/09/2021 et expertisée le 25/08/2021
Etat Mauvais
Ecart 21
Pour 110000 euros

[L'Exécution de Maximilien=> Manet, Française]
Achetée le 15/09/2021 et expertisée le 01/08/2021
Etat Mauvais
Ecart 45
Pour 250000 euros
---> Doit être expertisée

[Portrait d'Émile Zola=> Manet, Française]
Achetée le 15/09/2021 et expertisée le 15/06/2021
Etat Moyen
Ecart 92
Pour 95000 euros

Valeur : 1505000 euros. Ecart : 495000 euros

SALLE : Néerlandaise. Montant d'assurance : 2000000
Il y a 4 oeuvre(s)

[Tournesols dans un vase=> Van Gogh, Néerlandaise]
Achetée le 15/09/2021 et expertisée le 15/07/2021
Etat Mauvais
Ecart 62
Pour 750000 euros
---> Doit être expertisée

[Champ de ble avec cypres=> Van Gogh, Néerlandaise]
Prêtée par Metropolitan Museum pour 365 jours

[Autoportrait=> Van Gogh, Néerlandaise]
Achetée le 15/09/2021 et expertisée le 30/08/2021
Etat Mauvais
Ecart 16
Pour 1000000 euros

[Le Champ de blé aux iris=> Van Gogh, Néerlandaise]
Achetée le 15/09/2021 et expertisée le 15/07/2021
Etat Moyen
Ecart 62
Pour 95000 euros

Valeur : 1845000 euros. Ecart : 155000 euros

Tests des indexeurs

Tests des prédicats & tris

```
#region TESTS DES INDEXEURS

// -- Sur le RANG
Oeuvre uneOeuvre = francaise[1];
if (uneOeuvre != null)
    Console.WriteLine(uneOeuvre.GetNomOeuvre() + " TROUVEE");
else
    Console.WriteLine("Oeuvre non TROUVEE");

uneOeuvre = francaise[10];
if (uneOeuvre != null)
    Console.WriteLine(uneOeuvre.GetNomOeuvre() + " TROUVEE");
else
    Console.WriteLine("Oeuvre non TROUVEE");

// -- Sur le PRIX
Console.WriteLine();
Console.WriteLine("*** Oeuvres de la salle française de prix <= 500000 Euros");
List<Oeuvre> lesOeuvres = francaise[500000F];
if (lesOeuvres != null)
    foreach (Oeuvre o in lesOeuvres)
        Console.WriteLine(o.ToString());

#endregion
```

```
// -- Sur l'ARTISTE
Console.WriteLine();
Console.WriteLine("*** Oeuvres de MONET dans la salle française ");
lesOeuvres = francaise[monet];
if (lesOeuvres != null)
    foreach (Oeuvre o in lesOeuvres)
        Console.WriteLine(o.ToString());

// -- Sur le NOM de l'ARTISTE
Console.WriteLine();
Console.WriteLine("*** Oeuvres de MANET dans la salle française ");
lesOeuvres = francaise[manet.GetNomArtiste()];
if (lesOeuvres.Count != 0)
    foreach (Oeuvre o in lesOeuvres)
        Console.WriteLine(o.ToString());

Console.WriteLine();
Console.WriteLine("*** Oeuvres de DAVOLIO dans la salle française");
lesOeuvres = francaise["Davolio"];
if (lesOeuvres.Count != 0)
    foreach (Oeuvre o in lesOeuvres)
        Console.WriteLine(o.ToString());

#endregion
```

```
#region TEST PREDICATS ET TRIS

// Méthode "FIND".
// Renvoie le PREMIER élément de la collection pour
// lequel le prédicat est VRAI...
Console.WriteLine("\n\n*** Recherche des OEUVRES de VAN GOGH ***");
Predicats.nomArtiste = "Van Gogh";
List<Oeuvre> résultat = neerlandaise[Predicats.nomArtiste].FindAll(Predicats.RechercheOeuvresArtiste);
if (résultat.Count != 0)
    foreach (Oeuvre o in résultat)
        Console.WriteLine(o.ToString());
else
    Console.WriteLine("Aucune oeuvre trouvée");

// Utilisation de la méthode "Sort" pour TRIER
// les éléments de la collection d'OEUVRE suivant le NOM
lesOeuvres = new List<Oeuvre>() { o1, o2, o3, o4, o5, o6, o7, o8, o9, o10, o11, o12, o13 };

lesOeuvres.Sort(Predicats.ComparerOeuvresParNom);
Console.WriteLine("\n\n*** TRI par NOM ***");
foreach (Oeuvre o in lesOeuvres)
    Console.WriteLine(o.ToString());

Console.WriteLine();

// o5, o6 et o11 sont des oeuvres prêtées
lesOeuvres = new List<Oeuvre>() { o1, o2, o3, o4, o7, o8, o9, o10, o12, o13 };
lesOeuvres.Sort(Predicats.ComparerOeuvresParPrix);
Console.WriteLine("\n\n*** TRI par Prix ***");
foreach (Oeuvre o in lesOeuvres)
    Console.WriteLine(o.ToString());

#endregion

Console.ReadKey();
Console.Clear();
```

Résultats attendus :

```
Au bord de l'eau TROUVEE
Oeuvre non TROUVEE

*** Oeuvres de la salle française de prix <= 500000 Euros
[Le Déjeuner sur l'herbe=> Monet, Française]
Achetée le 15/09/2021 et expertisée le 15/07/2021
Etat Moyen
Ecart 62
Pour 500000 euros

[Au bord de l'eau=> Monet, Française]
Achetée le 15/09/2021 et expertisée le 15/05/2021
Etat Bon
Ecart 123
Pour 350000 euros

[La Plage à Honfleur=> Monet, Française]
Achetée le 15/09/2021 et expertisée le 01/07/2021
Etat Mauvais
Ecart 76
Pour 90000 euros
--> Doit être expertisée

[Quai du Louvre=> Monet, Française]
Achetée le 15/09/2021 et expertisée le 15/07/2021
Etat Mauvais
Ecart 62
Pour 110000 euros
--> Doit être expertisée

[Le petit Lange=> Monet, Française]
Achetée le 15/09/2021 et expertisée le 25/08/2021
Etat Mauvais
Ecart 21
Pour 110000 euros
```

```
[L'Exécution de Maximilien=> Manet, Française]
Achetée le 15/09/2021 et expertisée le 01/08/2021
Etat Mauvais
Ecart 45
Pour 250000 euros
--> Doit être expertisée

[Au bord de l'eau=> Monet, Française]
Achetée le 15/09/2021 et expertisée le 15/05/2021
Etat Bon
Ecart 123
Pour 350000 euros

[Le Déjeuner sur l'herbe=> Monet, Française]
Achetée le 15/09/2021 et expertisée le 15/07/2021
Etat Moyen
Ecart 62
Pour 500000 euros

[Tournesols dans un vase=> Van Gogh, Neerlandaise]
Achetée le 15/09/2021 et expertisée le 15/07/2021
Etat Mauvais
Ecart 62
Pour 750000 euros
--> Doit être expertisée

[Autoportrait=> Van Gogh, Neerlandaise]
Achetée le 15/09/2021 et expertisée le 30/08/2021
Etat Mauvais
Ecart 16
Pour 1000000 euros
```

```

*** OEUVRES A EXPERTISER (parmi l'ensemble des oeuvres de
Française

    [La Plage à Honfleur=> Monet, Française]
    Achetée le 15/09/2021 et expertisée le 01/07/2021
    Etat Mauvais
    Ecart 76
    Pour 90000 euros
    ---> Doit être expertisée

    [Quai du Louvre=> Monet, Française]
    Achetée le 15/09/2021 et expertisée le 15/07/2021
    Etat Mauvais
    Ecart 62
    Pour 110000 euros
    ---> Doit être expertisée

    [L'Exécution de Maximilien=> Monet, Française]
    Achetée le 15/09/2021 et expertisée le 01/08/2021
    Etat Mauvais
    Ecart 45
    Pour 250000 euros
    ---> Doit être expertisée

Neerlandaise

    [Tournesols dans un vase=> Van Gogh, Neerlandaise]
    Achetée le 15/09/2021 et expertisée le 15/07/2021
    Etat Mauvais
    Ecart 62
    Pour 750000 euros
    ---> Doit être expertisée

```

```

*** TRI par Prix ***
    [La Plage à Honfleur=> Monet, Française]
    Achetée le 15/09/2021 et expertisée le 01/07/2021
    Etat Mauvais
    Ecart 76
    Pour 90000 euros
    ---> Doit être expertisée

    [Portrait d'Émile Zola=> Monet, Française]
    Achetée le 15/09/2021 et expertisée le 15/06/2021
    Etat Moyen
    Ecart 92
    Pour 95000 euros

    [Le Champ de blé aux iris=> Van Gogh, Neerlandaise]
    Achetée le 15/09/2021 et expertisée le 15/07/2021
    Etat Moyen
    Ecart 62
    Pour 95000 euros

    [Quai du Louvre=> Monet, Française]
    Achetée le 15/09/2021 et expertisée le 15/07/2021
    Etat Mauvais
    Ecart 62
    Pour 110000 euros
    ---> Doit être expertisée

    [Le petit Lange=> Monet, Française]
    Achetée le 15/09/2021 et expertisée le 25/08/2021
    Etat Mauvais
    Ecart 21
    Pour 110000 euros

```

```

    [L'Exécution de Maximilien=> Monet, Française]
    Achetée le 15/09/2021 et expertisée le 01/08/2021
    Etat Mauvais
    Ecart 45
    Pour 250000 euros
    ---> Doit être expertisée

    [Au bord de l'eau=> Monet, Française]
    Achetée le 15/09/2021 et expertisée le 15/05/2021
    Etat Bon
    Ecart 123
    Pour 350000 euros

    [Le Déjeuner sur l'herbe=> Monet, Française]
    Achetée le 15/09/2021 et expertisée le 15/07/2021
    Etat Moyen
    Ecart 62
    Pour 500000 euros

    [Tournesols dans un vase=> Van Gogh, Neerlandaise]
    Achetée le 15/09/2021 et expertisée le 15/07/2021
    Etat Mauvais
    Ecart 62
    Pour 750000 euros
    ---> Doit être expertisée

    [Autoportrait=> Van Gogh, Neerlandaise]
    Achetée le 15/09/2021 et expertisée le 30/08/2021
    Etat Mauvais
    Ecart 16
    Pour 1000000 euros

```

Puis, avoir écrit le code permettant de tester la méthode `public Dictionary<Salle, List<Oeuvre_Achetee>> getLesExpertisesAFAire()` de la classe MUSEE :

```

*** OEUVRES A EXPERTISER (parmi l'ensemble des oeuvres de ci
Française

    [La Plage à Honfleur=> Monet, Française]
    Achetée le 15/09/2021 et expertisée le 01/07/2021
    Etat Mauvais
    Ecart 76
    Pour 90000 euros
    ---> Doit être expertisée

    [Quai du Louvre=> Monet, Française]
    Achetée le 15/09/2021 et expertisée le 15/07/2021
    Etat Mauvais
    Ecart 62
    Pour 110000 euros
    ---> Doit être expertisée

    [L'Exécution de Maximilien=> Monet, Française]
    Achetée le 15/09/2021 et expertisée le 01/08/2021
    Etat Mauvais
    Ecart 45
    Pour 250000 euros
    ---> Doit être expertisée

Neerlandaise

    [Tournesols dans un vase=> Van Gogh, Neerlandaise]
    Achetée le 15/09/2021 et expertisée le 15/07/2021
    Etat Mauvais
    Ecart 62
    Pour 750000 euros
    ---> Doit être expertisée

```