TRAVAIL PRATIQUE SYNTHÈSE

CONTEXTE

Le but de ce travail est de créer un programme permettant d'afficher le contenu d'une bibliothèque musicale. La bibliothèque est composée d'albums. Chaque album contient une liste de chanson et un ou plusieurs artistes.

DÉROULEMENT DE L'EXÉCUTION

Le programme **Musique.exe** distribué avec cet énoncé sert de référence comme résultat à atteindre.

LANCEMENT DU PROGRAMME

Le programme accepte comme argument de ligne de commande le nom des fichiers contenant les données de la bibliothèque. Le programme lit et charge en mémoire l'information de tous les fichiers fournis. Une fois le chargement terminé, il affiche des statistiques. Par exemple :

```
================================================
Chargement terminé
     2 Fichiers lus: musique1.txt, musique2.txt
   135 Albums chargés
  1389 Interprétations chargées
================================================
```

Il est possible que les fichiers contiennent de l'information corrompue. Le programme doit correctement traiter cette information et ignorer les lignes contenant des erreurs. Il est aussi possible qu'un nom de fichier fourni soit incorrect. Dans ce cas, des statistiques supplémentaires sont affichées. Par exemple :

```
================================================
Chargement terminé
     2 Fichiers lus: format.txt, erreurs_et_cas_limites.txt
    10 Albums chargés
    17 Interprétations chargées
    14 Lignes en erreur
     1 Fichier introuvable: patate.txt
================================================
```

Par la suite, le programme affiche le menu principal.

```
================================================================================
= Bibliothèque musicale                                                        =
================================================================================

 A) Afficher les albums
 R) Afficher les artistes
 C) Afficher les chansons
 Q) Quitter

> _
```

Option « A » : Afficher les albums

Cette option permet premièrement de choisir les critères d'affichage de la liste d'albums.

```
================================================================================
= Affichage des Albums                                                         =
================================================================================

 1) Choisir l'ordre de tri [A à Z]
 *) Afficher toute la liste
 Lettre) Afficher les items dont le nom débute par la lettre donnée
 2) Rechercher
 3) Retour au menu principal

> _
```

- L'option 1 permet de déterminer l'ordre d'affichage des albums : ordre alphabétique ou ordre alphabétique inverse. Le choix courant est affiché dans l'item du menu.

```
================================================================================
= Sélection du tri des Albums                                                  =
================================================================================

 A) A à Z
 Z) Z à A

> _
```

Cet ordre sera appliqué pour tous les affichages d'albums, tant qu'il ne sera pas changé.

- L'option '*' affiche la liste de tous les albums.
- Une lettre affiche la liste des albums dont la première lettre du nom est la lettre donnée.
- L'option 2 demande à l'utilisateur une chaine de caractère à rechercher. Seuls les albums dont le nom contient cette chaine seront affichés. La recherche n'est pas sensible à la casse.

Selon l'option choisie (liste complète, lettre de début ou recherche), la liste des albums correspondants est affichée, dans l'ordre choisi par l'option 1. Par exemple :

```
================================================================================
= Albums [A à Z] [Débutant par 'A'] (3)                                        =
================================================================================

   1: A Canadian Blues Rendez-Vous
   2: Alone And Acoustic
   3: An Evening Of Acoustic Music

 R) Retour

> _
```

Le programme n'affiche que 20 albums à la fois. Si la liste en contient plus, le programme offre l'option de continuer à la page suivante. Si une page suivante est affichée, le programme permet de naviguer à la page précédente.

```
 P) Page précédente
 S) Page suivante
 R) Retour

> _
```

Lorsqu'une liste d'albums est affichée, en plus des choix indiqués par le menu ('**P**', '**S**' ou '**R**'), il est possible de choisir le numéro d'un album dans la liste. Le numéro doit correspondre à une valeur actuellement affichée. Dans ce cas, les détails de l'album choisi sont affichés. Par exemple, dans la liste précédente, si le nombre « 2 » est choisi :

```
================================================================================
= Album                                                                        =
================================================================================

Alone And Acoustic
  Blues - 1991
  Artistes: Buddy Guy
            Junior Wells

Chansons
--------
  1. Give Me My Coat And Shoes  (3:51)
  2. Big Boat  (5:13)
  3. Diggin' My Potatoes  (4:28)
```

La page affiche le nom, le genre, la date, le ou les artistes et la liste d'interprétations. Les interprétations sont numérotées à partir de 1 et doivent être dans le même ordre que dans le fichier de données. Pour chaque interprétation, la durée est présentée en format « **Minutes:Secondes** ».

Si un album n'a pas d'artiste associé, l'indication « *Artistes variés* » est affichée. De plus, si une interprétation comporte un ou plusieurs artistes, ils sont affichés avec l'interprétation. Par exemple :

```
================================================================================
= Album                                                                        =
================================================================================

Bandits - Bande originale du film
  Rock/Pop - 2001
  Artistes variés

Chansons
--------
  1. Gallows Pole  (4:11)  [ Jimmy Page, Robert Plant ]
  2. Tweedle Dee & Tweedle Dum  (4:46)  [ Bob Dylan ]
  3. Holding Out For A Hero  (4:40)  [ Bonnie Tyler ]
```

Après l'affichage des détails d'un album, le programme retourne à la liste.

Option « R » : Afficher les artistes

Cette option permet premièrement de choisir les critères d'affichage de la liste d'artistes. Ce menu est identique à celui des albums, les options sont les mêmes avec la même signification.

Selon l'option choisie (liste complète, lettre de début ou recherche), la liste des artistes correspondants est affichée, dans l'ordre choisi par l'option 1. Par exemple :

```
================================================================================
= Artistes [Z à A] [Débutant par 'S'] (3)                                      =
================================================================================

   1: Sting
   2: Sex Pistols
   3: Sam Taylor

 R) Retour

> _
```

Comme pour les albums, le programme n'affiche que 20 artistes à la fois. Si la liste en contient plus, le programme offre aussi l'option de continuer à la page suivante et/ou revenir à la page précédente.

Lorsque le numéro d'un artiste est sélectionné, le programme affiche les détails de cet artiste. Par exemple :

```
================================================================================
= Artiste                                                                      =
================================================================================

Sting

Albums
------
  All This Time
  Brand New Day
  Nothing Like The Sun

Chansons
--------
  1. Ain't No Sunshine [Compilation Jazz]
  2. Nice Work If You Can Get It [The Glory Of Gershwin]
  3. Send Your Love [Awesome Mix, Volume 1]
```

La page affiche le nom de l'artiste.

Si l'artiste est associé à un ou plusieurs albums, ils sont affichés en ordre alphabétique.

Si l'artiste est associé à une ou plusieurs interprétations, elles sont affichées en ordre alphabétique. Les interprétations sont numérotées à partir de 1. Pour chaque interprétation, le nom de l'album sur lequel elle se retrouve est indiqué. La durée n'est pas affichée.

Après l'affichage des détails d'un artiste, le programme retourne à la liste.

Option « C » : Afficher les chansons

Cette option permet premièrement de choisir les critères d'affichage de la liste de chansons. Ce menu est identique à celui des albums et celui des artistes, les options sont les mêmes avec la même signification.

Selon l'option choisie (liste complète, lettre de début ou recherche), la liste des chansons correspondantes est affichée, dans l'ordre choisi par l'option 1. Par exemple :

```
================================================================================
= Chansons [A à Z] [Recherche "away"] (3)                                      =
================================================================================

   1: Don't Give Me Away
   2: They Can't Take That Away From Me
   3: Wishing The Rain Away

 R) Retour

>
```

Comme pour les albums et artistes, le programme n'affiche que 20 chansons à la fois. Si la liste en contient plus, le programme offre aussi l'option de continuer à la page suivante et/ou revenir à la page précédente.

Lorsque le numéro d'une chanson est sélectionné, le programme affiche les détails de cette chanson. Par exemple :

```
================================================================================
= Chanson                                                                      =
================================================================================

They Can't Take That Away From Me

Disponible sur les albums
-------------------------
  1. The Best Of Frank Sinatra  (2:42)
  2. Swing When You're Winning  (3:07)  [ Rupert Everett ]
  3. The Glory Of Gershwin  (3:16)  [ Lisa Stansfield ]
```

La page affiche le nom de la chanson et la liste des albums sur lesquels se retrouve une interprétation de la chanson.

Les interprétations sont numérotées à partir de 1. L'ordre des interprétations n'est pas défini. Pour chaque interprétation, la durée est présentée en format « **`Minutes:Secondes`** ». Si une interprétation comporte un ou plusieurs artistes, ils sont affichés avec l'interprétation.

Après l'affichage des détails d'une chanson, le programme retourne à la liste.

Fichiers de données

Chaque ligne du fichier représente un enregistrement qui peut être de 2 types : album ou interprétation. Le premier caractère indique le type de l'enregistrement : « **`A`** » pour album, « **`I`** » pour interprétation. Les champs d'un enregistrement sont séparés par des points-virgules « `;` ».

Le format d'un album est :

- Nom (chaine de caractères)
- Année (entier)
- Genre (chaine de caractères)
- Artistes (liste optionnelle de chaines de caractères).

Le format d'une interprétation est :

- Nom (chaine de caractères)
- Durée en secondes (entier)
- Artistes (liste optionnelle de chaines de caractères).

Il peut y avoir des espaces avant et après chaque champ qui doivent être éliminées lors de la lecture.

Si le premier caractère non blanc d'une ligne est « **#** », cette ligne est un commentaire qui est ignoré. Les lignes vides ou ne contenant que des espaces sont aussi ignorées.

Si une ligne est en erreur, car elle ne respecte pas le format attendu, elle est ignorée.

Les fichiers **format.txt** et **erreurs_et_cas_limites.txt** fournis donnent plus d'information et des exemples sur le format des données.

Les fichiers **musique1.txt** et **musique2.txt** fournis sont à utiliser pour tester l'exécution de votre programme.

IMPORTANT : Les fichiers ne sont lus qu'**une seule fois**, au lancement du programme. Toute l'information est chargée en mémoire.

CONSIGNES

Créez une nouvelle solution avec un projet nommé **`TPSynthese`**, et ajoutez-y les fichiers fournis avec cet énoncé.

EXIGENCES

- Le programme est écrit en C#, en mode console
- Le programme respecte les principes de la programmation orientée objet :
  - Encapsulation (données regroupées dans un objet)
    Il ne doit avoir **aucun** attribut public, **même** s'ils sont **`readonly`**.
  - Abstraction (présente une interface en cachant les détails d'implémentation)
    Les propriétés ne doivent être publiquement modifiables que si nécessaire.
  - Héritage (regroupement des fonctionnalités communes).
  - Polymorphisme (accès aux fonctionnalités spécifiques à travers un type de base).
- Assurez-vous d'avoir une solution qui compile même si elle n'est pas 100 % fonctionnelle.
- Le programme doit valider toutes les entrées de l'utilisateur et résister à toute donnée erronée.
- Les attributs non modifiables après l'initialisation doivent être qualifiés de **`readonly`**.
- Utilisez des constantes, lorsqu'applicables.
- N'utilisez le niveau d'accès **`protected`** que lorsque ça a du sens.

CONTRAINTES

Le programme doit **au minimum** respecter le diagramme suivant :

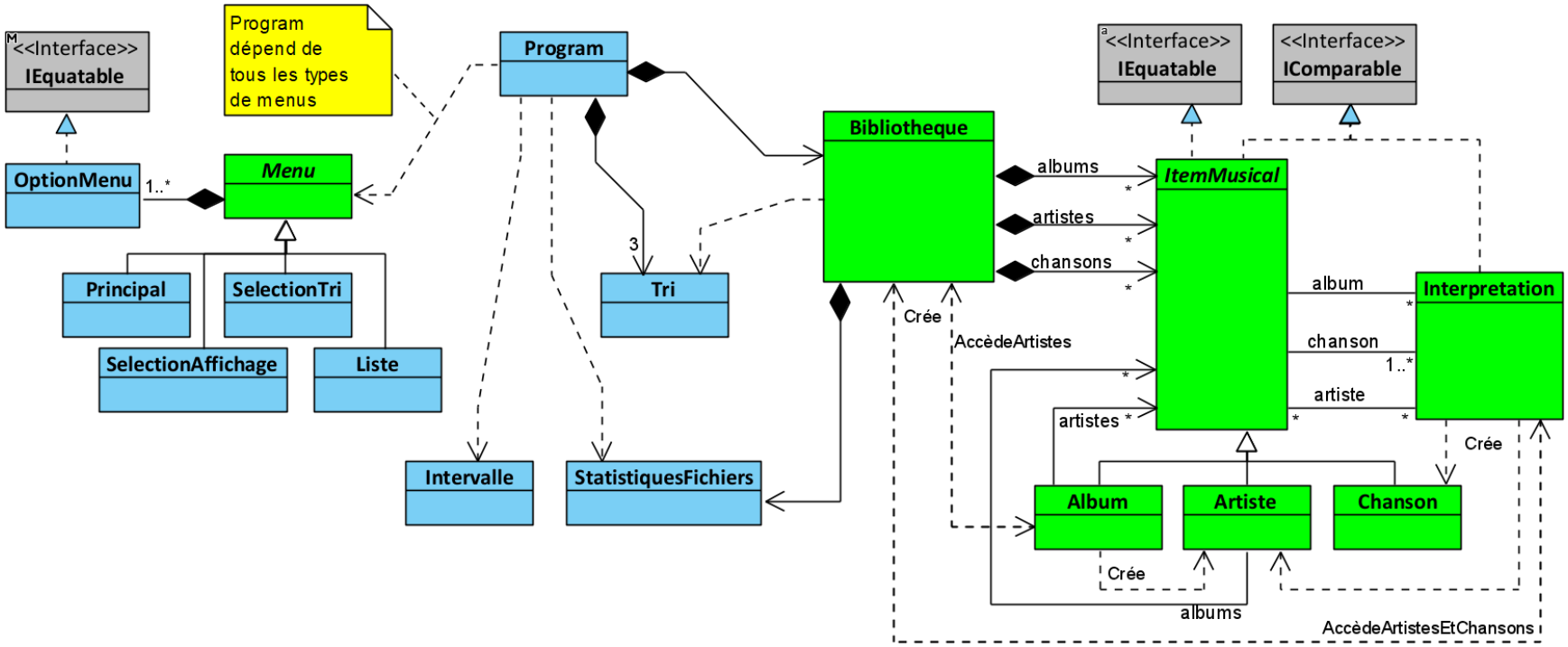

Diagramme de classes minimal du programme

- Les classes en bleu vous sont fournies avec cet énoncé. Vous **ne devez pas modifier** ces classes.
- Les classes en vert et leurs relations telles que présentées dans le diagramme sont à définir. Elles doivent s'adapter à la logique de la classe `Program` et des autres classes fournies.
- Les classes `Program` et `Menu` sont les seules à pouvoir accéder à la `Console`. Les autres classes ne doivent ni lire ni écrire à la console.

Classes fournies

Program

La classe principale de l'application, elle contient la méthode **`Main`**. Elle gère le déroulement général de l'application pour toutes les options des différents menus.

Menu.OptionMenu

Représente une option dans un menu. Utilisée par la classe **`Menu`** pour toutes les options disponibles. Permet d'afficher l'option du menu, ainsi que de valider l'existence d'une option dans un menu.

*Menu.Principal, Menu.SelectionAffichage, Menu.SelectionTri, Menu.Liste (fichier **Menus.cs**)*

Spécialisations de la classe **`Menu`**. Chaque classe correspond à l'un des menus de l'application. La classe **`Menu`** à définir doit correspondre à l'implémentation de ces classes.

StatistiquesFichiers

Conteneur de statistiques. Lors de la lecture des fichiers de données, les méthodes pour incrémenter les statistiques ou ajouter un fichier doivent être appelées correctement. Cette classe gère l'affichage des statistiques.

Tri

Une petite classe simple pour indiquer l'ordre du tri, de A à Z, ou de Z à A. Trois attributs de ce type sont contenus dans la classe **`Program`** (un pour chaque type d'item musical) et donnés à la classe **`Bibliotheque`** pour la génération des listes d'items.

Intervalle

Utilisée par la classe **`Program`** pour gérer la navigation entre les pages lors de l'affichage des listes. Complètement interne à la classe **`Program`**, vous n'avez pas à utiliser cette classe.

Classes et énumérations requises

TypeItem

Une énumération qui définit les trois types d'items manipulés par le programme : **`Album`**, **`Artiste`** et **`Chanson`**.

Menu

La classe de base pour tous les menus définis dans le fichier **Menus.cs**.

- Permet d'afficher un titre dans le haut de la console (méthode **`AfficherTitre`**).
- Permet d'afficher un menu et de demander le choix de l'utilisateur (méthode **`Afficher`**).
  - Vérifie le choix de l'utilisateur et le redemande tant qu'il n'est pas valide. Elle ne retourne qu'un choix valide.
  - Certains menus acceptent des valeurs numériques (différentes de zéro). Si l'une de ces valeurs est donnée, elle est retournée.
  - Certains menus acceptent des caractères autres que des chiffres. Si l'un de ces caractères est donné, il est mis dans le paramètre et la valeur zéro est retournée.
  - Accepte des lettres minuscules ou majuscules, mais retourne toujours des lettres majuscules.

Interpretation

La représentation d'une chanson sur un album. Comme une chanson portant le même nom peut se retrouver plusieurs fois dans la bibliothèque (sur le même album, ou des albums différents), un seul objet de type **`Chanson`** sera créé afin de n'apparaitre qu'une seule fois dans la liste de chansons. Un objet de type **`Interpretation`** est créé pour chaque interprétation lue des fichiers de données.

- Une interprétation est créée par la bibliothèque lors de la lecture des fichiers.
- Contient les détails provenant du fichier de données puisque ceux-ci sont propres à l'interprétation et peuvent variés pour une même chanson.
- Contient un item musical qui est l'album sur lequel l'interprétation se retrouve.
- Contient un item musical qui est la chanson de laquelle l'objet est une interprétation.
  - Lors de sa création, vérifie si la bibliothèque contient déjà la chanson. Si oui, conserve une référence sur cette chanson.
  - Sinon, elle crée une nouvelle chanson et l'insère dans la bibliothèque.
- Contient une liste d'items musicaux qui sont les artistes, optionnels, associés à l'interprétation.
  - Lors de sa création, si un ou des artistes sont associés à l'interprétation, pour chacun des artistes, vérifie si la bibliothèque contient déjà l'artiste. Si oui, conserve une référence sur cet artiste.
  - Sinon, elle crée un nouvel artiste et l'insère dans la bibliothèque.
- Il doit être possible de trier une liste d'interprétation en ordre alphabétique.

ItemMusical

Classe de base pour tous les types d'items manipulés par le programme : **`Album`**, **`Artiste`** et **`Chanson`**.

Après la création d'un item spécifique (album, artiste ou chanson), le programme ne manipule que des objets de type **`ItemMusical`**. Tout le traitement doit être générique, peu importe le type réel.

- Contient une liste d'interprétation. Ceci est commun à tous les items spécifiques.
- Il doit être possible de trier une liste d'items musicaux en ordre alphabétique.
- Il doit être possible de retrouver un item musical dans une liste s'il porte le même nom qu'un autre item.

Album

Spécialisation d'**`ItemMusical`** pour représenter un album lu du fichier. Un objet de ce type est créé pour chaque album lu des fichiers de données.

- Un album est créé par la bibliothèque lors de la lecture des fichiers.
- Contient une liste d'items musicaux qui sont les artistes, optionnels, associés à l'album.
  - Lors de sa création, si un ou des artistes sont associés à l'album, pour chacun des artistes, vérifie si la bibliothèque contient déjà l'artiste. Si oui, conserve une référence sur cet artiste.
  - Sinon, elle crée un nouvel artiste et l'insère dans la bibliothèque.

Artiste

Spécialisation d'**`ItemMusical`** pour représenter un artiste.

- Un artiste est créé par un album ou par une interprétation lors de sa création, s'il n'existe pas déjà.
- Contient une liste d'items musicaux qui sont les albums auxquels l'artiste est associé.

Chanson

Spécialisation d'**`ItemMusical`** pour représenter une chanson.

- Une chanson est créée par une interprétation lors de sa création, si elle n'existe pas déjà.

Bibliotheque

Conteneur de toute l'information chargée des fichiers de données. Un seul objet de cette classe existe comme attribut de la classe **`Program`**.

- Contient une liste d'items musicaux qui sont tous les albums connus de l'application.
- Contient une liste d'items musicaux qui sont tous les artistes connus de l'application.
- Contient une liste d'items musicaux qui sont toutes les chansons connues de l'application.
- Le constructeur lit tous les fichiers donnés au lancement du programme.
  - Pour chaque ligne valide débutant par « **`A`** », un album est créé et ajouté à la liste des albums.
  - Pour chaque ligne valide débutant par « **`I`** », une interprétation est créée. La bibliothèque ne contient pas de liste de toutes les interprétations.

L'information doit être représentée par des objets. Par exemple, un album ne doit pas avoir une liste de chaines de caractères pour les noms des artistes, mais bien une liste d'objets de type **`Artiste`**.

Les classes représentant les données de la bibliothèque (album, artiste, chanson et interprétation) ne doivent être instanciées qu'une seule fois pour chaque donnée. Par exemple, si un artiste est associé à plusieurs albums et/ou chansons, un seul objet de type **`Artiste`** doit exister pour le représenter.

PONDÉRATION

Ce travail compte pour **20** % de la note finale

- Détermination correcte des attributs, des méthodes des classes et des structures de données 20 %
- Application judicieuse des principes d'encapsulation et d'héritage 15 %
- Implémentation correcte des classes dans le programme 25 %
- Notation claire et pertinente de commentaires en nombre approprié dans le code informatique 10 %
- Respect systématique des normes de programmation 10 %
- Fonctionnement correct du programme 20 %

PÉNALITÉS

Français : Jusqu'à 10 % de pénalité

Retard : Pénalité de 10 % par jour de retard (incluant les fins de semaine)

Plagiat : Ceci est un travail individuel. Application de la Politique institutionnelle sur la fraude le plagiat et la tricherie pour *toutes les personnes* impliquées.

REMISE

Vous devez compresser les fichiers sources (**.cs**) de votre programme, **sans** inclure les fichiers fournis. N'incluez aucun autre fichier.

Vous devez remettre le fichier dans LÉA, dans la rubrique Travaux, pour le TP Synthèse.

Vous devez remettre votre travail avant **12:30** le **lundi 17 mai 2021**.

RAPPEL DES NORMES DE NOMENCLATURE ET DE DOCUMENTATION

- Chaque classe est déclarée dans son propre fichier.
- Les variables, incluant les paramètres de méthode, sont définies en camelCase. Les attributs de classes sont préfixés d'un souligné (_). Tout le reste est défini en PascalCase.
- Dans la déclaration d'une classe, les éléments publics sont déclarés en premier, puis les éléments protégés (protected) (si applicable), et finalement les éléments privés.
- Le mot clé **`private`** est explicitement indiqué.
- Les blocs d'instructions sont toujours délimités par des accolades.

```
// Faire
if (laCondition)
{
   AfficherMessage();
}

// Ne pas faire
if (laCondition)
   AfficherMessage();
```

Exception : Dans une instruction **`switch`**, il n'est pas nécessaire d'ajouter des accolades à chaque bloc d'instructions d'un **`case`**.

- Une seule déclaration de variable est présente sur une ligne.

```
// Faire
int dividende;
int diviseur;

// Ne pas faire
int dividende, diviseur;
```

- Ne pas laisser de blocs d'instructions vides, sauf dans le cas d'un bloc **`catch`** qui ignore silencieusement les erreurs (ajoutez un commentaire pour expliquer).

```
// Faire
if (! laCondition)
{
   ...
}

// Ne pas faire
if (laCondition)
{
}
else
{
   ...
}
```

COMMENTAIRES

- Chaque déclaration de classe comprend un commentaire en format *CodeDoc* décrivant la classe.
- Chaque méthode comprend un commentaire en format *CodeDoc* décrivant la fonctionnalité de la méthode et, lorsqu'applicable, les paramètres, la valeur de retour, et les exceptions pouvant être lancées par la méthode.
- Le code contient des commentaires explicatifs pertinents en quantité appropriée.
- Si vous utilisez une notion non vue en classe, vous devez ajouter un commentaire expliquant **en détail** son fonctionnement.
- Si vous vous inspirez de code trouvé, vous devez ajouter un commentaire pour citer la source.