Partage d'Expériences en Pédagogie Innovante

UNIVERSITÉ Clermont Auvergne

IUT DE CLERMONT-FD
Aurillac · Le Puy-en-Velay · Clermont-Ferrand
UNIVERSITÉ Clermont Auvergne

# Journée PEPI

## Jeudi 27 juin 2019

## Évaluations blanches en mode projet

Marc Chevaldonné

Amphithéâtre de l'IADT, 51 bd François Mitterrand

Partage d'Expériences en Pédagogie Innovante

IUT DE CLERMONT-FD
UNIVERSITÉ Clermont Auvergne

UNIVERSITÉ Clermont Auvergne

## Contexte

- **DUT Informatique 1A**, 2A, Licence Pro Informatique
- Fin de 1A (fin mars à fin juin)
- 7 semaines ; par étudiant, par semaine :
  - 2h CM (136 étudiants)
  - **4h TD** (36 étudiants)
  - **4h TP** (18 étudiants)
  - **Travail à la maison**
- Bases de la conception orientée objets, Interface Homme Machine

## Constat

- **Les étudiants**
  - ne travaillent **pas** tous **en même temps**, **pas** tous **à la même vitesse**, **pas** tous **régulièrement**
  - abandonnent vite si le début n’est pas compris
  - peuvent ne pas s’intéresser à cause du contexte
  - n’osent pas poser de questions devant leurs camarades
  - ne connaissent pas leur niveau avant les retours de l’évaluation finale…
  - … qu’ils ne lisent pas, car ils arrivent trop tard

## Constat

- Les enseignants
  - un cours qui a peu de chances d’être lu et est vite caduc
  - les examens sont :
    - longs à préparer
    - longs à corriger
    - le feedback est trop tardif

    ➡sentiment de sanctionner plus que d’enseigner

Partage d'Expériences en Pédagogie Innovante

IUT DE CLERMONT-FD
Aurillac · Le Puy-en-Velay · Clermont-Ferrand
UNIVERSITÉ Clermont Auvergne

UNIVERSITÉ Clermont Auvergne

## Mode projet

1. les étudiants **choisissent leur thème et leur sujet**
2. ils reçoivent une **fiche détaillée des compétences** à acquérir (production : documents + logiciel)

# Fiche de compétences

Ajouté par Marc CHEVALDONNE il y a moins d'une minute

*Rappel* :

- ne rendez qu'un seul document (pdf de préférence), contenant l'intégralité des schémas, diagrammes, descriptions pour les 3 modules,
- ne rendez qu'une seule solution faites de plusieurs projets et ressources pour vos programmes
- il y a 2 notes par module :
  - une partie écrite (appelée « documents »)
  - une partie développement (appelée « programmation »)

- une évaluation blanche n'est qu'indicative : elle ne comptera pas dans la moyenne. Le soin apporté aux corrections n'est pas le même que pour l'évaluation finale et les notes blanches ne sont qu'un aperçu de votre travail à un instant t.
- Critères d'évaluation pour chaque note :
  (Note : le barème n'est pas définitif et très susceptible d'évoluer ; il n'est donné qu'à titre indicatif)

*Bilan*
Au total : ***/120***

## Objets 2 : Conception et Programmation Orientées Objets (C#, .NET)

### Documents : /20

- diagramme de paquetage [sur 2 points]
- diagramme de classes [sur 8 points]
- diagramme de séquence (sur quelques cas particuliers) [sur 2 points]
- description écrite de l'architecture (dont patrons de conception, dépendances…) [sur 8 points]
- Note : chaque diagramme doit être accompagné de notes et d'une description écrite.

### Programmation : /20

- bases (classes, structures, instances, …) [sur 2 points]
- abstraction (héritage, interfaces, polymorphisme) [sur 3 points]
- collections simples (tableaux, listes…) [sur 2 points]
- collections avancées (dictionnaires) [sur 2 points]
- encapsulation [sur 5 points]
- tests (fonctionnels et/ou unitaires) [sur 4 points]
- LINQ [sur 1 point]
- évènements (cf. module IHM) [sur 1 point]

## IHM : Interface Homme-Machine (XAML, WPF)

**Documents : /20**

- description du contexte [sur 4 points]
- sketchs [sur 4 points]
- storyboards [sur 4 points]
- diagramme de cas d'utilisation [sur 5 points]
- considérations ergonomiques [sur 2 points]
- prise en compte de l'accessibilité [sur 1 point]

**Programmation : /20**

- XAML :
  - répartition dans l'espace (layout des vues et usercontrols) [sur 2 points]
  - utilisation des controls (vues et usercontrols) [sur 1 point]
  - ressources, styles [sur 2 points]
  - DataTemplate (locaux et globaux) [sur 2 points]

- boucle Model <-> View
  - gestion d'évènements sur la vue [sur 2 points]
  - gestion d'évènements depuis le métier (notifications) [sur 2 points]
  - DataBinding (sur le Master) [sur 2 points]
  - DataBinding (sur le Detail) [sur 2 points]
  - DataBinding sur les UserControl + Dependency Property [sur 2 points]

- gestion du Master-Detail [sur 3 points]

## Projet Tuteuré S2

**Documents : /20**

- diagramme de paquetage mettant en avant la partie persistance [sur 2 points]
- diagramme de classes mettant en avant la partie persistance [sur 4 points]
- diagramme de classes sur votre (vos) partie(s) ajoutée(s) [sur 4 points]
- vidéo de 1 à 3 minute(s) du projet [sur 10 points]

**Programmation : /20** ¶

- persistance (XML, JSON, BDD, WebService, ...) [sur 3 points]

Partage d'Expériences en Pédagogie Innovante

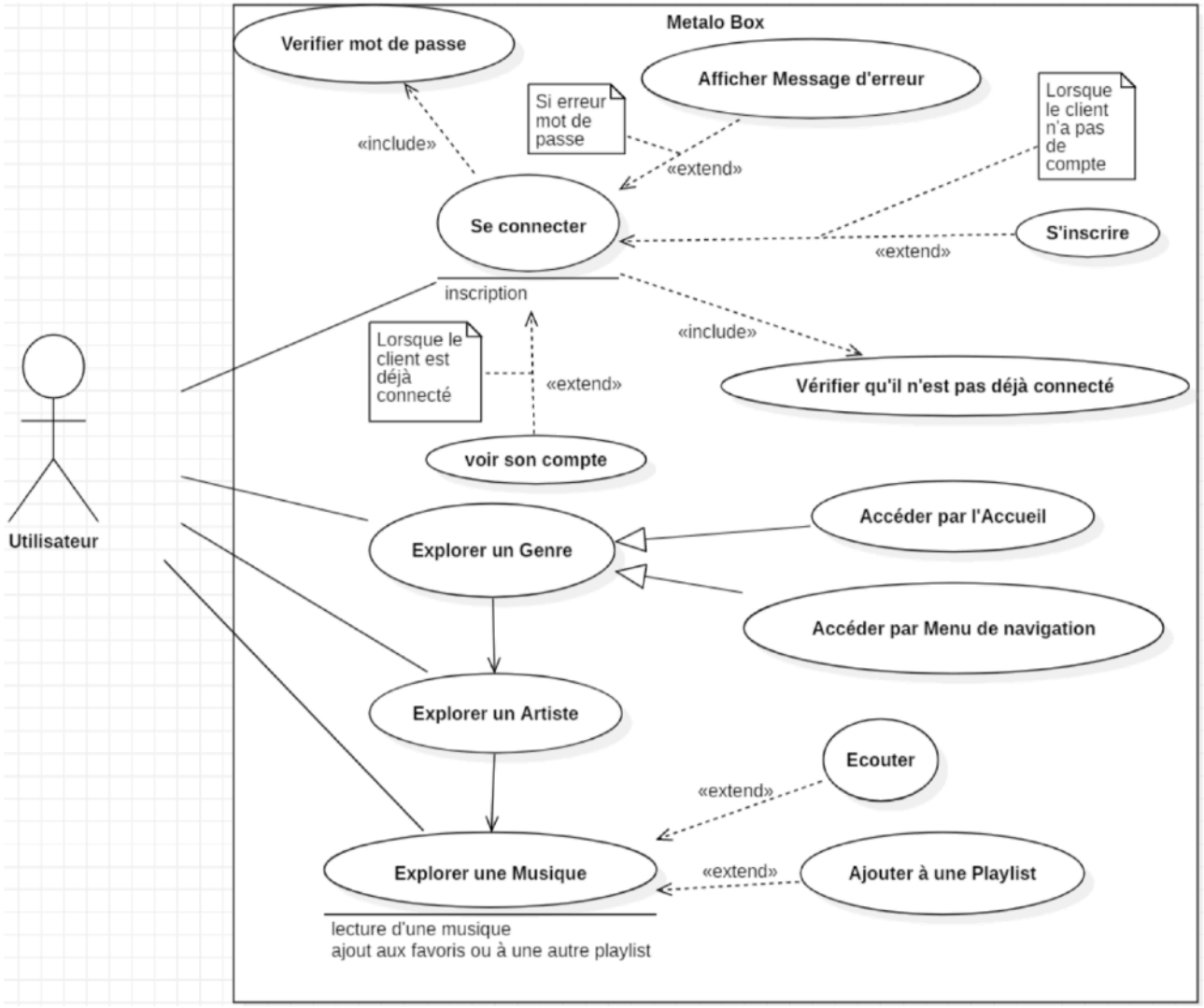
IUT DE CLERMONT-FD
Aurillac · Le Puy-en-Velay · Clermont-Ferrand
UNIVERSITÉ Clermont Auvergne

UNIVERSITÉ Clermont Auvergne

## Mode projet

1. les étudiants **choisissent leur thème et leur sujet**
2. ils reçoivent une **fiche détaillée des compétences** à acquérir (production : documents + logiciel)

Diagramme cas d'utilisation



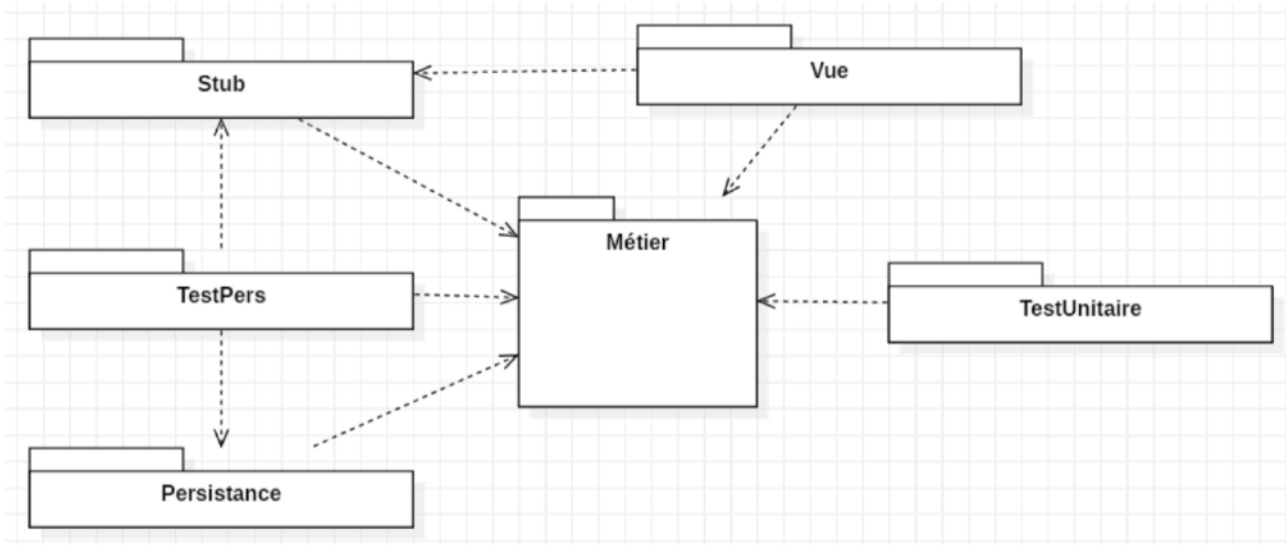

Notre application n'a qu'un acteur principal : l'Utilisateur, qui utilise comme système notre Application. Sur cette application, il peut exécuter de nombreuses actions représentées par des cas d'utilisation (les ronds).

La première action possible est la **connexion** de l'utilisateur. Lorsqu'il se connecte cela inclut que son mot de passe soit vérifié d'où la relation d'*inclusion*. S'il se trompe, alors l'application affiche un message d'erreur. Ce cas ne se déroulant que sous certaines conditions (ici l'utilisateur doit s'être

-Stub et XML : Stub sert à tester la persistance et XML est le moyen choisi pour sauvegarder. (voir plus en détail dans la description de l'architecture)

-IPersistance : Interface que doivent implémenter tous les moyens de sauvegarde : ici Stub et XML.

Description Diagramme de paquetage:

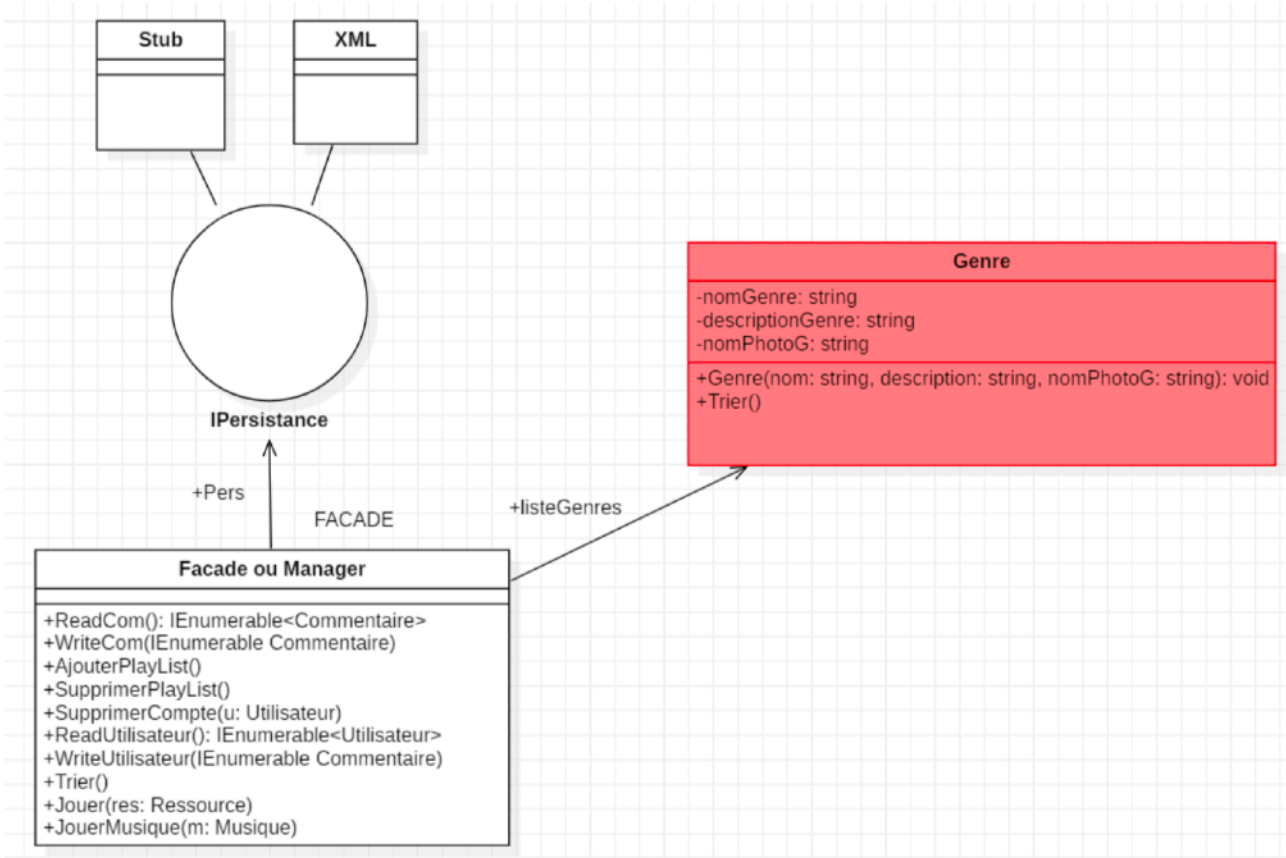



L'application est constituée de plusieurs parties :

-la vue, une partie visible où se trouvent toutes les fenêtres en XAML
-le métier, constitué d'une partie invisible : les classes C# et la façade
-le test unitaire permet de vérifier que les classes C# fonctionnent comme souhaité
-la persistance XML sert à sauvegarder
-le Stub permet de simuler une sauvegarde sans vraiment sauvegarder
-le TesttPers permet de tester rapidement le Stub, en vérifiant que ses méthodes fonctionnent et utilise aussi la Persistance Xml pour les sauvegardes.

Autre exemple, pour charger ou sauvegarder, on appellera la façade qui à son tour appellera la IPersistance qu'on lui a passée en paramètre (ici cela peut être soit un Stub, soit un XML car ce sont les 2 classes qui implémentent IPersistance). La persistance choisie s'exécutera alors pour faire ce que lui a demandé la façade, et lui renverra ce dont elle a besoin.

2)Strategie et persistance




Pour la persistance nous utilisons le patron de conception Stratégie. Le principe consiste à crée une interface IPersistance qui possède toutes les méthodes communes aux différents moyens de persistance. Nous en utilisons 2 dans l'application. Ainsi on pourra changer de moyen changer de moyen de persistance quand on veut et en un seul mot, simplement en changeant le IPersistance donné en paramètre lorsque l'on crée la façade :

Partage d'Expériences en Pédagogie Innovante

IUT DE CLERMONT-FD
Aurillac · Le Puy-en-Velay · Clermont-Ferrand
UNIVERSITÉ Clermont Auvergne

UNIVERSITÉ Clermont Auvergne

## Mode projet

1. les étudiants **choisissent leur thème**
2. ils reçoivent une **fiche détaillée des compétences** à acquérir (production : documents + logiciel)
3. CM : **conseils** d'organisation, **démonstrations** live
4. TD/TP : **questions / réponses** + manipulations

Partage d'Expériences en Pédagogie Innovante

IUT DE CLERMONT-FD
UNIVERSITÉ Clermont Auvergne

UNIVERSITÉ Clermont Auvergne

## Mode projet

1. les étudiants **choisissent leur thème**
2. ils reçoivent une **fiche détaillée des compétences** à acquérir (production : documents + logiciel)
3. CM : **conseils** d'organisation, **démonstrations** live
4. TD/TP : **questions / réponses** + manipulations
5. outils : forge, subversion/git

# root

URL : https://forge.clermont-universite.fr/svn/musxplorer | Statistiques | Révision :

| Nom | Taille | Révision | Âge | Auteur | Commentaire |
|---|---|---|---|---|---|
| Assemblage | | 91 | 3 jours | Mathilde PAPILLON | Correction de certaines fautes dans les fichier... |
| Classe | | 91 | 3 jours | Mathilde PAPILLON | Correction de certaines fautes dans les fichier... |
| Contexte | | 92 | 3 jours | Mathilde PAPILLON | Suppression du Dossier "Test" qui était inutile... |
| MenuApp | | 91 | 3 jours | Mathilde PAPILLON | Correction de certaines fautes dans les fichier... |
| Stub | | 84 | 7 jours | Antoine FERRAND | DOCUMENTS TOUS RASSEMBLES DANS LE PDF -ajouts d... |
| TestStub | | 91 | 3 jours | Mathilde PAPILLON | Correction de certaines fautes dans les fichier... |
| TestUnitaire | | 84 | 7 jours | Antoine FERRAND | DOCUMENTS TOUS RASSEMBLES DANS LE PDF -ajouts d... |
| Xml | | 91 | 3 jours | Mathilde PAPILLON | Correction de certaines fautes dans les fichier... |
| MenuApp.sln | 3,419 ko | 59 | 28 jours | Mathilde PAPILLON | Ajout de protocole d'égalité sur certaine class... |
| PROJET PDF.pdf | 2,054 Mo | 92 | 3 jours | Mathilde PAPILLON | Suppression du Dossier "Test" qui était inutile... |
| VidéoMetalobox.mp4 | 44,445 Mo | 91 | 3 jours | Mathilde PAPILLON | Correction de certaines fautes dans les fichier... |
| blackm.jpg | 72,287 ko | 13 | 3 mois | Antoine FERRAND | Ajout sur la fenêtre "genres" +image + scrollVi... |
| desktop.ini | 402 octet | 3 | 3 mois | Antoine FERRAND | |
| projet final.odt | 4,595 Mo | 92 | 3 jours | Mathilde PAPILLON | Suppression du Dossier "Test" qui était inutile... |

## Dernières révisions

| # | Date | Auteur | Commentaire |
|---|---|---|---|
| 92 | 21/06/2019 09:34 | Mathilde PAPILLON | Suppression du Dossier "Test" qui était inutile<br>Encore quelques modifs dans les fichiers annexes mais rien d'important |

# Révisions

Révision : OK

| # | Date | Auteur | Commentaire |
|---|---|---|---|
| 92 | 21/06/2019 09:34 | Mathilde PAPILLON | Suppression du Dossier "Test" qui était inutile<br>Encore quelques modifs dans les fichiers annexes mais rien d'important |
| 91 | 21/06/2019 09:24 | Mathilde PAPILLON | Correction de certaines fautes dans les fichiers annexes (PDF, Explications Diagrammes ..)<br>Rajout de quelques commentaires dans le code<br>suppression d'un fichier .txt inutile et d'une image<br>Ajout de la vidéo |
| 90 | 20/06/2019 19:57 | Antoine FERRAND | -changements timer pour player |
| 89 | 20/06/2019 14:27 | Antoine FERRAND | -Erreur enfin corrigé (le 2ème typeof j'avais mis la classe Artiste au lieu du UC)<br>-Groupe.xaml renommé en Artistes.xaml car plus cohérent<br>-Le binding pour aller sur Artistes ne marche toujours pas cependant<br>-Le binding d'après pour naviguer une fois sur Artistes marche |
| 88 | 18/06/2019 23:23 | Antoine FERRAND | -Zoom sur la partie Ecouter |
| 87 | 18/06/2019 21:04 | Mathilde PAPILLON | Modification des fenêtres Inscription et Connexion<br>Navigation de la page Connexion à Compte quand l'utilisateur se connecte<br>Changement pour le diagramme de paquetage et modification dans le fichier "Explications Diagramme" + ajout dans le fichier libreOffice pour le PDF... |
| 86 | 18/06/2019 19:54 | Antoine FERRAND | -modifs UML<br>-Trier la liste d'Artiste Alphabétiquement (croissant ou décroissant) |
| 85 | 17/06/2019 22:57 | Antoine FERRAND | -Erreur pour afficher la page Artiste |
| 84 | 17/06/2019 02:58 | Antoine FERRAND | DOCUMENTS TOUS RASSEMBLES DANS LE PDF<br>-ajouts de contenu pour descriptions (notamment les schémas) et notamment les sketchs balsamiq à l'intérieur.<br>-début du binding pour afficher la page artiste, mais ne marche pas encore |
| 83 | 16/06/2019 22:47 | Antoine FERRAND | -Diagramme de cas complet avec description et tout<br>-Ergonomie et accessibilité |
| 82 | 16/06/2019 21:17 | Mathilde PAPILLON | Ajout des persistances Artiste et Musique<br>(Ajout aussi de Groupe mais en commentaire, au cas où qu'il faille l'utiliser |
| 81 | 16/06/2019 16:20 | Antoine FERRAND | -Les Artistes s'affichent correctement<br>-Il ont des musiques mais on n'accède pas encore à la page Groupe donc on peut pas les voir ni rien encore |

```
179     179                 serializer.WriteObject(writer, gen7);
180     180                 serializer.WriteObject(writer, gen8);
        180                 serializer.WriteObject(writer, desGenres);
181     181             }
182     182
183     183
184     184
```

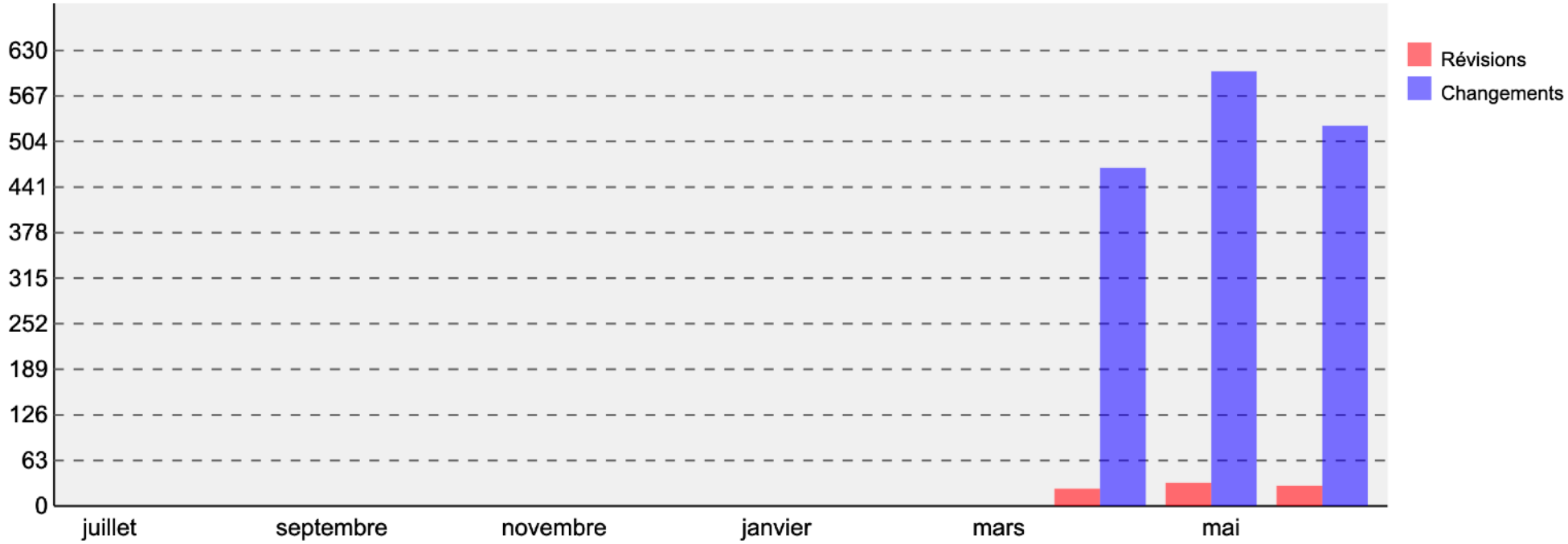
**Assemblage/Genre.cs**

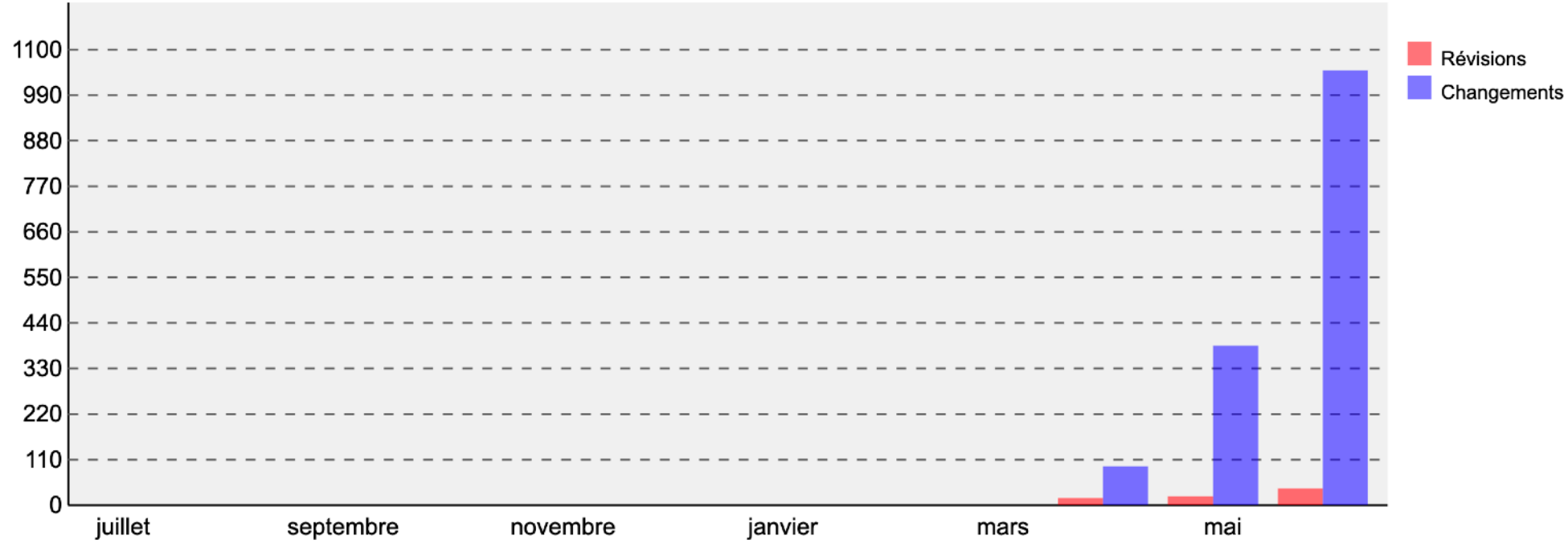
```
64      64          [DataMember]
65      65          public string NomPhoto { get; set; }
66      66
        67          [DataMember]
        68          public string ImageBouton { get; set; }
        69
67      70          private List<Artiste> listeGroupe = new List<Artiste>();
68      71
69              -   public Genre(string nom, string description, string nomPhotoG)
        72          public Genre(string nom, string description, string nomPhotoG, string imageBouton)
70      73          {
71      74              Nom = nom;
72      75              Description = description;
73      76              NomPhoto = nomPhotoG;
        77              ImageBouton = imageBouton;
74      78          }
75      79
76      80          public void AjouterArtiste(params Artiste[] artistes)
...     ...
85      89
86      90          public override string ToString()
87      91          {
88              -       return $"{Nom}, {Description} {NomPhoto})";
        92              return $"{Nom}, {Description} {NomPhoto} {ImageBouton}";
89      93          }
90      94
91      95
```

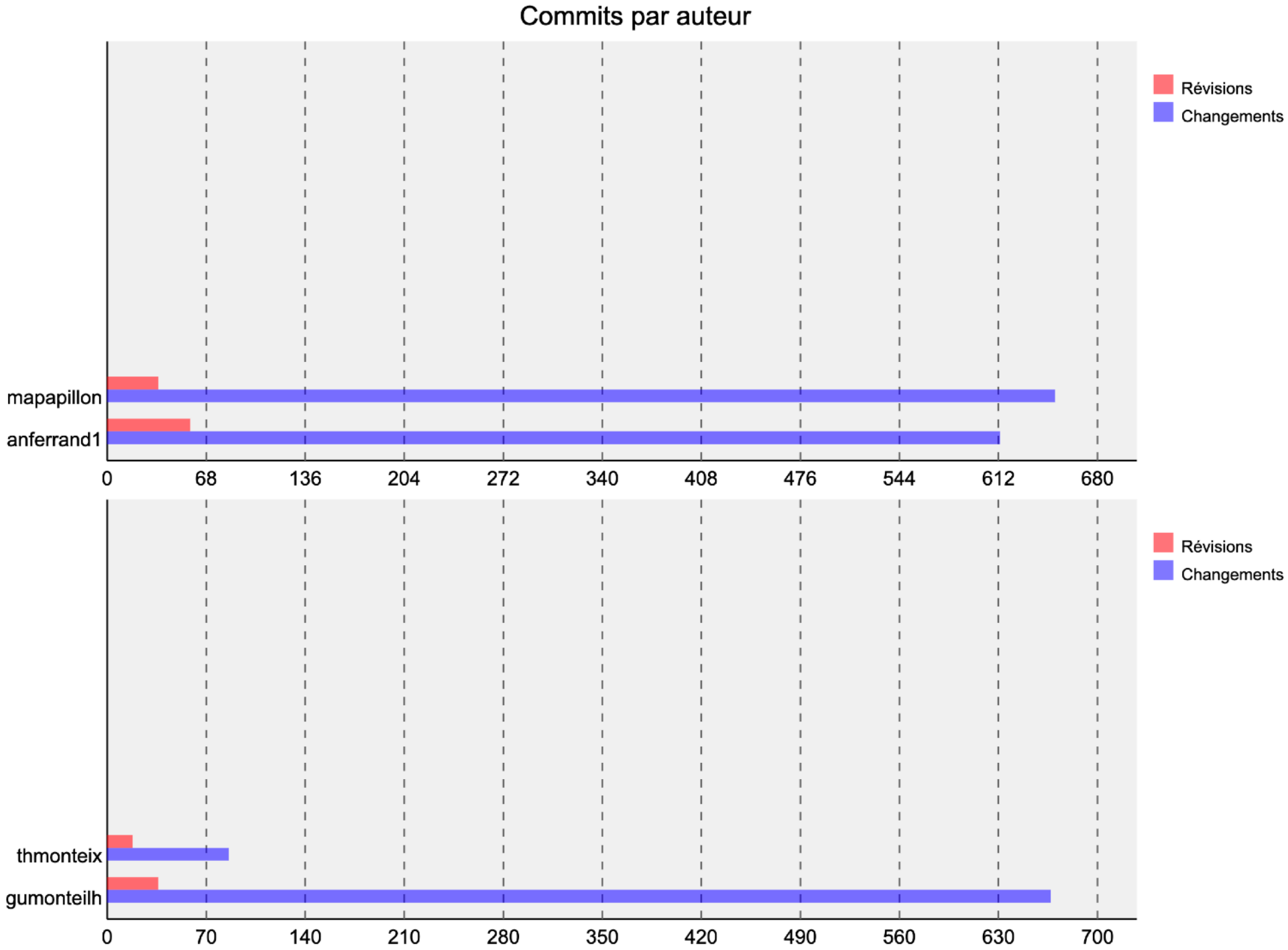
**Assemblage/Groupe.cs**

```
16  16          public string Description { get; set; }
17  17          public DateTime DateCreation { get; set; }
18  18          public string Pays { get; set; }
19              private List<Personne> listePers = new List<Personne>();
```

## Commits par mois




Commits par auteur
Révisions
Changements
mapapillon
anferrand1
0
68
136
204
272
340
408
476
544
612
680
Révisions
Changements
thmonteix
gumonteilh
0
70
140
210
280
350
420
490
560
630
700

Partage d'Expériences en Pédagogie Innovante

IUT DE CLERMONT-FD
UNIVERSITÉ Clermont Auvergne

UNIVERSITÉ Clermont Auvergne

## Mode projet

1. les étudiants **choisissent leur thème**
2. ils reçoivent une **fiche détaillée des compétences** à acquérir (production : documents + logiciel)
3. CM : **conseils** d'organisation, **démonstrations** live
4. TD/TP : **questions / réponses** + manipulations
5. outils : forge, subversion/git
6. ressources : nombreux exemples sur Moodle et gitlab

mchSamples .NET Framework

Project
Details
Activity
Releases
Cycle Analytics
Repository
Issues 0
Merge Requests 0
CI / CD
Operations
Wiki
Snippets
Settings
Collapse sidebar

| | | |
|---|---|---|
| ex_023_001_IEnumerator_ex1 | samples of .NET framework | 2 months ago |
| ex_023_002_IEnumerator_ex2 | samples of .NET framework | 2 months ago |
| ex_023_003_IEnumerator_ex3 | samples of .NET framework | 2 months ago |
| ex_023_004_ArrayClass | samples of .NET framework | 2 months ago |
| ex_023_005_Queue | samples of .NET framework | 2 months ago |
| ex_023_006_Stack | samples of .NET framework | 2 months ago |
| ex_023_007_LinkedList | samples of .NET framework | 2 months ago |
| ex_023_008_List | samples of .NET framework | 2 months ago |
| ex_023_009_HashSet_and_SortedSet | samples of .NET framework | 2 months ago |
| ex_023_010_EqualityProtocoleOnReferences | samples of .NET framework | 2 months ago |
| ex_023_011_EqualityProtocoleOnValues | samples of .NET framework | 2 months ago |
| ex_023_012_EqualityComparer | samples of .NET framework | 2 months ago |
| ex_023_013_OrderComparisonProtocole | samples of .NET framework | 2 months ago |
| ex_023_014_Dictionary | samples of .NET framework | 2 months ago |
| ex_023_015_DictionaryCustomType | samples of .NET framework | 2 months ago |
| ex_023_016_ReadOnlyCollection | samples of .NET framework | 2 months ago |
| ex_023_017_Deep_Read_only_collections_dll | samples of .NET framework | 2 months ago |
| ex_023_017_Deep_Read_only_collections_exe | samples of .NET framework | 2 months ago |
| ex_023_018_ReadOnlyDictionnaries | samples of .NET framework | 2 months ago |
| ex_023_019_ReadOnlyDictionary_and_values_dll | samples of .NET framework | 2 months ago |

Partage d'Expériences en Pédagogie Innovante

IUT DE CLERMONT-FD
Aurillac · Le Puy-en-Velay · Clermont-Ferrand
UNIVERSITÉ Clermont Auvergne

UNIVERSITÉ Clermont Auvergne

# Evaluations blanches : retours fréquents, quantifiés, commentés

Il s'agit d'une indication

Ne compte pas dans la moyenne !

## Evaluation blanche du 03 mai 2019 - 09h56

Ajouté par Marc CHEVALDONNE il y a environ un mois

*Rappel* :

- ne rendez qu'un seul document (pdf de préférence), contenant l'intégralité des schémas, diagrammes, descriptions pour les 3 modules,
- ne rendez qu'une seule solution faites de plusieurs projets et ressources pour vos programmes
- il y a 2 notes par module :
  - une partie écrite (appelée « documents »)
  - une partie développement (appelée « programmation »)

- une évaluation blanche n'est qu'indicative : elle ne comptera pas dans la moyenne. Le soin apporté aux corrections n'est pas le même que pour l'évaluation finale et les notes blanches ne sont qu'un aperçu de votre travail à un instant t.
- Critères d'évaluation pour chaque note :
  (Note : le barème n'est pas définitif et très susceptible d'évoluer ; il n'est donné qu'à titre indicatif)

*Bilan*
Au total : ***19/120***

### Objets 2 : Conception et Programmation Orientées Objets (C#, .NET)

**Documents : 3/20**

- diagramme de paquetage [sur 2 points]
- diagramme de classes [sur 8 points]
  ***Très bien pour la forme à part quelques coquilles (l'association entre Album et Musique, et entre Musique et Playlist par exemple). L'UML est bien respecté.***
  ***Attention important : vos classes ne doivent pas s'afficher ; elles doivent rendre les données affichables accessibles aux vues et tests.***
  ***Améliorations possibles :***
  ***a. gestion des groupes de groupes***
  ***b. gestion des métiers par musique plutôt que par artiste pour permettre d'avoir une personne parfois musicienne, parfois productrice, parfois les deux, etc...***
  ***c. gestion des ressources pour permettre d'avoir aussi bien des fichiers audio en local que des liens youtube ou autre...***
  ***Il manque la description (-4)***
  ***=> 3/8***
- diagramme de séquence (sur quelques cas particuliers) [sur 2 points]
- description écrite de l'architecture (dont patrons de conception, dépendances...) [sur 8 points]
- Note : chaque diagramme doit être accompagné de notes et d'une description écrite

**Programmation : 3/20**

- bases (classes, structures, instances, …) [sur 2 points]
  ***OK pour l'écriture des classes.***
  ***Attention à l'écriture des propriétés (une propriété automatique ne doit pas être accompagnée d'un attribut, sinon, doublons).***
  ***=> 1/2***
- abstraction (héritage, interfaces, polymorphisme) [sur 3 points]
  ***Un premier héritage, mais pas encore exploité***
  ***=> 1/3***
- collections simples (tableaux, listes…) [sur 2 points]
  ***OK, mais les listes ne sont pas encore très exploitées et il manque les protocoles d'égalité.***
  ***=> 0,5/2***
- collections avancées (dictionnaires) [sur 2 points]
- encapsulation [sur 5 points]
  ***Pas encore d'encapsulation (setters publics, collection non protégée).***
  ***=> 0/5***
- tests (fonctionnels et/ou unitaires) [sur 4 points]
  ***oui mais... un peu court...***
  ***=> 0,5/4***
- LINQ [sur 1 point]
- évènements (cf. module IHM) [sur 1 point]

**Programmation : 7/20**

- bases (classes, structures, instances, …) [sur 2 points]
  ***OK pour l'écriture des classes.***
  ***~~Attention à l'écriture des propriétés (une propriété automatique ne doit pas être accompagnée d'un attribut, sinon, doublons).~~***
  ***Attention à la boucle infinie dans Genres.***
  ***A quoi sert le foreach dans Groupe ?***
  ***=> 1/2***
- abstraction (héritage, interfaces, polymorphisme) [sur 3 points]
  ***~~Un premier héritage, mais pas encore exploité~~***
  ***Bien compris pour la stratégie de persistance. Il faudrait maintenant faire en sorte qu'on puisse persister autre chose que des commentaires. Mais c'est un bon début.***
  ***L'héritage sur Artiste reste à exploiter.***
  ***=> 2/3***
- collections simples (tableaux, listes…) [sur 2 points]
  ***~~OK, mais les listes ne sont pas encore très exploitées et il manque les protocoles d'égalité.~~***
  ***Il manque de nombreux protocoles d'égalité. Celui qui est écrit est mal implémenté.***
  ***=> 0,5/2***
- collections avancées (dictionnaires) [sur 2 points]
  ***Pas encore utilisé***
  ***=> 0/2***
- encapsulation [sur 5 points]
  ***~~Pas encore d'encapsulation (setters publics, collection non protégée).~~***
  ***Des progrès aussi bien sur les propriétés que les collections. Effort à poursuivre.***
  ***=> 2,5/5***
- tests (fonctionnels et/ou unitaires) [sur 4 points]
  ***~~oui mais... un peu court...~~***
  ***C'est compris pour les tests unitaires, maintenant il faut les remplir.***
  ***L'application Console doit être une application pas une bibliothèque de classes. A compléter.***
  ***=> 1/4***
- LINQ [sur 1 point]
- évènements (cf. module IHM) [sur 1 point]

**Programmation : 9,5/20**

- bases (classes, structures, instances, …) [sur 2 points]
  ***OK pour l'écriture des classes.***
  ***~~Attention à la boucle infinie dans Genres.~~***
  ***A quoi sert le foreach dans Groupe ?***
  ***=> 1,5/2***
- abstraction (héritage, interfaces, polymorphisme) [sur 3 points]
  ***Bien compris pour la stratégie de persistance. Il faudrait maintenant faire en sorte qu'on puisse persister autre chose que des commentaires. Mais c'est un bon début.***
  ***L'héritage sur Artiste reste à exploiter.***
  ***=> 2/3***
- collections simples (tableaux, listes…) [sur 2 points]
  ***~~Il manque de nombreux protocoles d'égalité.~~ ~~Celui qui est écrit est mal implémenté.~~***
  ***De nombreux protocoles d'égalité sont mal implémentés.***
  ***=> 1/2***
- collections avancées (dictionnaires) [sur 2 points]
  ***Pas encore utilisées***
  ***=> 0/2***
- encapsulation [sur 5 points]
  ***~~Des progrès aussi bien sur les propriétés que les collections.~~ Effort à poursuivre.***
  ***=> 3,5/5***
- tests (fonctionnels et/ou unitaires) [sur 4 points]
  ***C'est compris pour les tests unitaires, maintenant il faut les remplir.***
  ***~~L'application Console doit être une application pas une bibliothèque de classes.~~ Test fonctionnel à compléter.***
  ***Le test fonctionnel ne fonctionne pas.***
  ***=> 1/4***
- LINQ [sur 1 point]
- évènements (cf. module IHM) [sur 1 point]
  ***Un PropertyChanged dans la façade.***
  ***=> 0,5/1***

**Programmation : 12,5/20**

- bases (classes, structures, instances, …) [sur 2 points]
  ***OK pour l'écriture des classes.***
  ***~~A quoi sert le foreach dans Groupe ?~~***
  ***=> 2/2***
- abstraction (héritage, interfaces, polymorphisme) [sur 3 points]
  ***Bien compris pour la stratégie de persistance. ~~Il faudrait maintenant faire en sorte qu'on puisse persister autre chose que des commentaires. Mais c'est un bon début.~~***
  ***L'héritage sur Artiste reste à exploiter.***
  ***=> 2/3***
- collections simples (tableaux, listes…) [sur 2 points]
  ***~~De nombreux protocoles d'égalité sont mal implémentés.~~***
  ***Il manque quelques protocoles d'égalité.***
  ***=> 1,5/2***
- collections avancées (dictionnaires) [sur 2 points]
  ***Pas encore utilisées***
  ***A quoi sert le dictionnaire dans Musique ?***
  ***=> 0/2***
- encapsulation [sur 5 points]
  ***Effort à poursuivre.***
  ***=> 4/5***
- tests (fonctionnels et/ou unitaires) [sur 4 points]
  ***C'est compris pour les tests unitaires, maintenant il faut les remplir.***
  ***~~Test fonctionnel à compléter.~~***
  ***~~Le test fonctionnel ne fonctionne pas.~~***
  ***Bien pour le test de persistance.***
  ***=> 2,5/4***
- LINQ [sur 1 point]
- évènements (cf. module IHM) [sur 1 point]
  ***Un PropertyChanged dans la façade.***
  ***=> 0,5/1***

## Evaluations blanches : retours fréquents, quantifiés, commentés

- **Pour les étudiants**
  - Permet de connaître son niveau par rapport aux attentes de l'enseignant
  - Faiblement commentés pour encourager les questions
  - Permet de travailler à la hauteur de ses ambitions, capacités, compétences
  - Force à approfondir
  - Projet d'envergure
  - 4 évaluations blanches par binôme

## Evaluations blanches : retours fréquents, quantifiés, commentés

- **Pour les enseignants**
  - 1-2 corrections par jour pendant 12 semaines
  - 20-40 minutes par correction
    - - de travail en amont
    - + de travail pendant
    - - de travail après
  - Corrections régulières => + de proximité et possibilité d'adapter aux compétences de chacun

## Et la note finale ?

- Évaluation du projet sur le modèle de l'évaluation blanche
- Court oral
  - Discussion avec l'étudiant
  - Prise en compte de la participation de l'étudiant
  - Vérification du niveau par rapport à celui du projet
- Vidéo à fournir
  https://mc01.dsi.uca.fr/videos/?video=MEDIA190624030343841

## Avantages / Inconvénients

- **Pour les étudiants**
  - TD/TP plus interactifs et adaptés à chaque étudiant
  - + d'implication
  - Projet + concret
  - Préféré par 34/35 étudiants
- **Pour les enseignants**
  - Changement de rythme
  - - monotone
  - - de travail avant et après
  - Il faut être disponible pendant
  - Il faut corriger régulièrement

## Perspectives d'améliorations

- CM en décalage + conseils a posteriori
  - anticipation du retard des étudiants
  - profiter des bénéfices des erreurs et des échecs
- capsules vidéos
- questions en direct et anonymes en CM et TD avec Wooclap