**Travail pratique # 5 :** Restructuration du code source

École polytechnique de Montréal

Trimestre : automne 2017

Équipier1 : Georges Louis 1880098

Équipier2 : Mazigh Ouanes XXXXXXX

Équipe numéro : 20

Date de remise (4-12-2017)

**E1) Une mauvaise odeur dans les attributs**

**1) Expliquez pourquoi c’est une odeur grave. [/2]**

La Classe « Emission.h » est une mauvaise odeur parce qu’elle a un grand nombre d’attributs. C’est une classe qui agit comme 2 sous classes.

**2) IdentiHiez le nom de la restructuration nécessaire pour enlever cette odeur du code. [/2]**

La restructuration nécessaire pour enlever cette odeur est l’extraction d’une classe. Cette dernière consiste à créer une nouvelle classe (Class Chaine) et déplacer les attributs et méthodes pertinents vers la nouvelle classe.

**3) IdentiHiez les méthodes qui seront impactées par ce changement. [/6]**

Les méthodes qui seront impactées par ce changement sont celles que notre nouvelle Class Chaine aura besoin et par la suite notre Class « Emission.h » va les perdes :

* Chaine(string, string, string); méthode ajoutée
* Chaine(); méthode ajoutée
* string getChaineName();
* string getChaineCodePostal();
* string getChaineAddress();

Donc, il faut faire un changement dans quelques méthodes de la Class « Emission.h ».  
Les méthodes qui seront modifiés sont les suivantes :

* void associerChaine(Chaine\*)
* Chaine\* getChaine(); méthode ajoutée

**Identifiez les attributs qu’il faut modifier ou déplacer. [/3]**

Les attributs qui seront déplacé de la classe « Emission.h » vers la Class Chaine sont :

• string chaineName;

• string chaineCodePostal;

• string chaineAddress;

Pour la classe Émission, il y a un seul attribut qui faut ajouter et qui est le suivant :

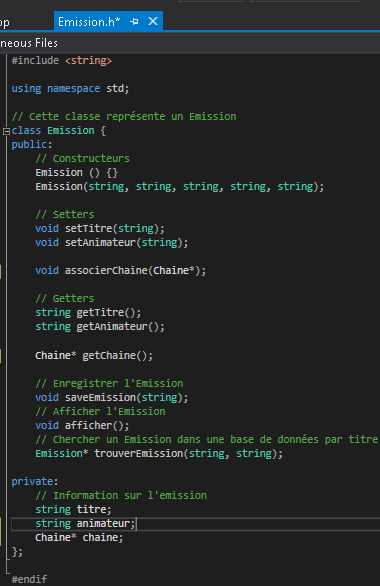
• Chaine\* chaine;

**Identifiez les étapes à suivre pour restructurer cette odeur. Utilisez le même format du tableau cidessous, dans lequel vous décomposez la restructuration globale en étapes plus simples [/5]**

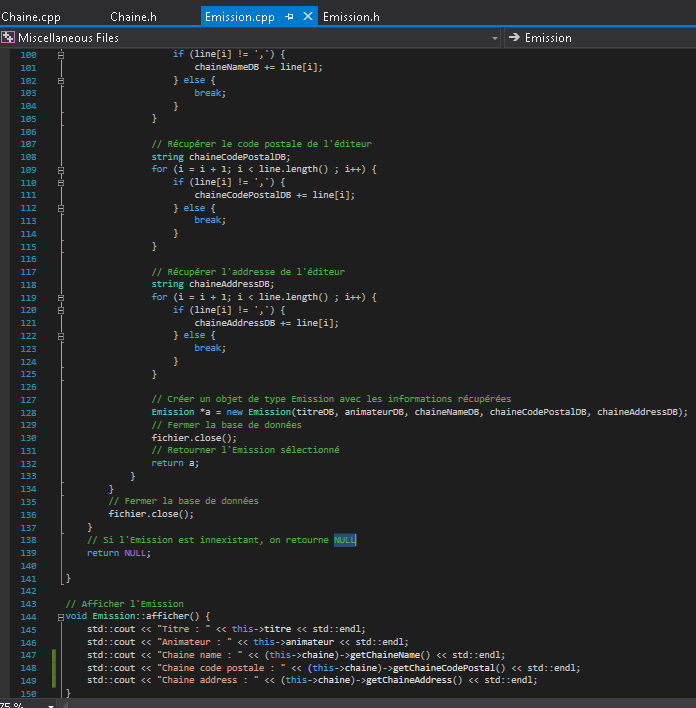
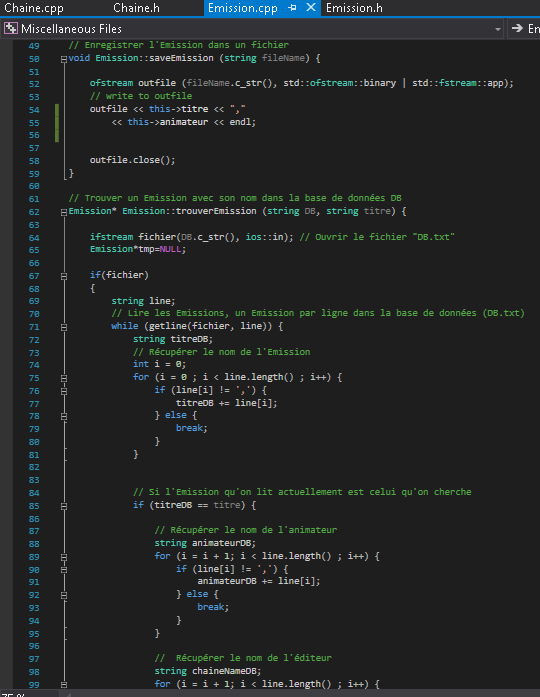
|  |  |
| --- | --- |
| Étape | Description |
| Créer une nouvelle classe Chaine | Créer un fichier Chaine.h et Chaine.cpp |
| Écrire le fichier Chaine.h | Transférer les définitions des méthodes et des attributs dans le fichier Chaine.h. |
| Écrire le fichier Chaine.cpp | Écrire les implémentations des méthodes et des attributs de la classe Chaine.h |
| Modifier la classe Emission.h | Enlever les définitions des méthodes et des attributs qui font parties de la classe Chaine.h |
| Modifier la classe Emission.cpp | Enlever les implémentations des méthods et des attributs qui font parties de la classe Chaine.cpp |

**Restructurez le code source en modiHiant/déplaçant les attributs de la question 4 et en modiHiant les méthodes impactées (question 3). (Notez que vous avez Étape Description le droit de créer une nouvelle classe en cas de besoin). Copiez la/les classe(s) modiHiée(s) dans le rapport (le Hichier header et cpp). [/6]**

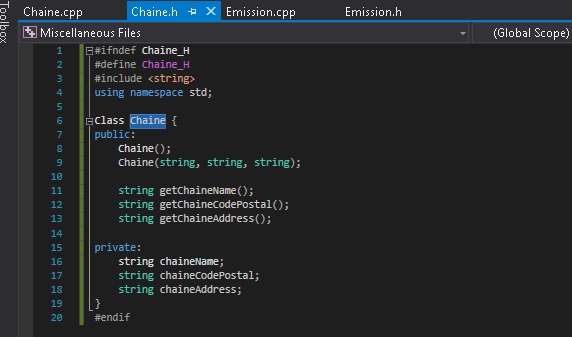
Emission.h

****

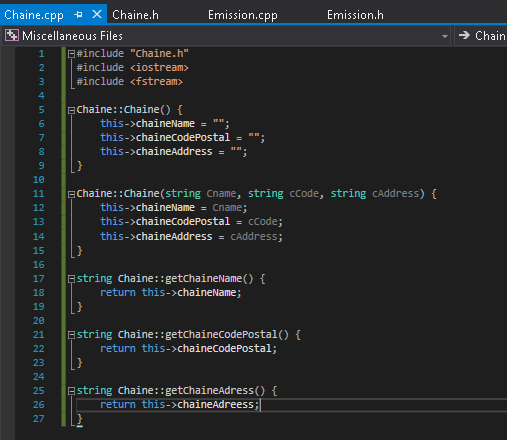
Emission.cpp



Chaine.h



Chaine.cpp



**Compilez et exécutez les tests unitaires fournis (/tests/…), veuillez ajouter des captures d’écrans des résultats de vos tests dans le rapport. [/4]**

**Faites un commit de votre code source, et une capture d’écran de ce commit dans le rapport (Le résultat de la commande « git add, commit, et push »). [/2]**

**E2) Une mauvaise odeur dans les méthodes [/20]**

**En examinant la méthode « TrouverEmission» de la classe « Emission.cpp » : Identifiez le nom des deux odeurs graves et expliquez pourquoi ce sont des odeurs graves. [/2]**

Les deux odeurs graves de la méthode « TrouverEmission » sont la duplication du même code et la méthode est trop longue. On peut créer une méthode qui fait le fonctionnement du code qui est dupliqué.

**Planifiez, étape par étape, comment restructurer cette odeur, dans le même format du tableau de l’exercice E1. [/6]**

|  |  |
| --- | --- |
| Étape | Description |
| Déterminer le code qui est dupliqué | Trouver les répétitions du même code dans la méthode “TrouverEmission” |
| Créer une nouvelle méthode | La nouvelle méthode créer doit faire le travaille du code qui est dupliqué |

**Restructurez le code source de cette méthode. Copiez dans le rapport le nouveau code de la méthode, ainsi que d’autres méthodes si vous en créez des nouvelles ou si vous modifiez d’autres méthodes dans cette restructuration. [/6]**

Nouvelle méthode créer :

String Emission ::FindInfo(string line) {

String information = “ ”;

For (int i = 0; i < line.length(); i++){

If(line[i] != ‘,’) {

Information += line[i];

}

Else {

Break;

}

}

return information;

}

// Trouver un Emission avec son nom dans la base de données DB

Emission\* Emission::trouverEmission (string DB, string titre) {

ifstream fichier(DB.c\_str(), ios::in); // Ouvrir le fichier "DB.txt"

Emission\*tmp=NULL;

if(fichier) {

string line;

// Lire les Emissions, un Emission par ligne dans la base de données (DB.txt)

while (getline(fichier, line)) {

// Récupérer le nom de l'Emission

int i = 0;

string titreDB = FindInfo(line);

// Si l'Emission qu'on lit actuellement est celui qu'on cherche

if (titreDB == titre) {

// Récupérer le nom de l'animateur

string animateurDB = FindInfo(line);

// Récupérer le nom de l'éditeur

string chaineNameDB = FindInfo(line);

// Récupérer le code postale de l'éditeur

string chaineCodePostalDB = FindInfo(line);

// Récupérer l'addresse de l'éditeur

string chaineAddressDB = FindInfo(line);

// Créer un objet de type Emission avec les informations récupérées

Emission \*a = new Emission(titreDB, animateurDB, chaineNameDB, chaineCodePostalDB, chaineAddressDB);

// Fermer la base de données

fichier.close();

// Retourner l'Emission sélectionné

return a;

}

}

// Fermer la base de données

fichier.close();

}

// Si l'Emission est innexistant, on retourne NULL

return NULL;

}

**Compilez et exécutez les tests unitaires fournis (/tests/…), veuillez ajouter des captures d’écrans des résultats de vos tests dans le rapport. [/4]**

**Faites un commit de votre code source, et une capture d’écran de ce commit dans le rapport. [/2]**

**E3) Utilisation des variables [/22]**

**Calculez le span, la durée de vie et la portée des variables « DBFile», «choix » et « emission» dans la méthode « main » dans « main.cpp ». Les lignes vides ne comptent pas ! [/6]**

**Interprétez les résultats, et trouvez la variable (parmi les trois citées en dessus) qui bénéHiciera le plus de la restructuration. [/2]**

**Proposez des restructurations pour améliorer l’utilisation de cette variable, en utilisant le même format du tableau de l’exercice E1. [/4]**

**Effectuez cette restructuration dans la méthode « main ». Faites une capture d’écran (ou copiez le code) de votre nouveau code source. [/4]**

**Compilez et testez manuellement (en exécutant le programme sur la ligne de commandes) les opérations (de l’opération 0 à 4) de la méthode « main », veuillez prendre des captures d’écrans de vos tests. [/4]**

**Faites un commit de votre code source, et une capture d’écran de ce commit dans le rapport. [/2]**

**E4) Questions de cours [/10]**

1. **Citez deux avantages de l’utilisation des APIs. Quel problème un développeur peut avoir en utilisant une API non stable ? [/1]**
2. **Citez deux librairies qui fournissent une API de bonne qualité. [/1]**
3. **Comme la plupart de vous ont déjà touché à la programmation orienté objet, citez des erreurs que vous avez faites. Et comment allez vous les éviter après la lecture de “les APIs et les classes” ? [/1]**
4. **Quelles sont les critères qu’on doit prendre en considération lors de l’implémentation d’une méthode ? [/1]**
5. **Pour faciliter la lecture de votre code, quels sont les pratiques que vous ne devez pas oublier ? [/1]**
6. **Quelle est la différence entre une “Cohésion fonctionnelle” et une “Cohésion de communication” ? [/1]**
7. **DéHinissez “patron de conception”, “anti-patron de conception” et “mauvaise odeur du code” et le lien entre eux ? [/2]**
8. **(BONUS) Nous sommes rendus à la Hin des TPs du LOG1000. Comment avez vous trouvez les TPs ? et d'après vous quels sont les changements qu’on doit apporter aux TPs ? [/2]**