

LOG1000- Ingénierie logicielle Processus de développement
d'un projet logiciel open source

Travail pratique #5

Stephanie Mansour (1935596)

Mariam Sarwat (1928777)

Section : 02

École Polytechnique de Montréal

23 novembre 2018

E1) Une mauvaise odeur dans les attributs

1. Il s'agit d'une odeur grave, car on se retrouve avec une classe trop grosse. En effet, comme indiqué, les attributs de cette classe jouent deux rôles différents et n'ont pas de lien entre eux. Certains attributs sont liés à des informations sur l'article (son nom et auteur), tandis que les autres sont liés à des informations sur l'éditeur de l'article (son nom, code postal et adresse). Par conséquent, nous devrions nous retrouver avec deux classes au lieu d'une seule et on fera une relation d'agrégation puisque l'article possède un éditeur.
2. Le nom de la restructuration nécessaire pour enlever cette odeur du code est l'extraction d'une classe. Il faudrait créer une nouvelle classe EditeurArticle et y placer les attributs et méthodes concernant l'information de l'éditeur de l'article. Dans la classe Article, on gardera les attributs et méthodes concernant l'information de l'article.
3. Les méthodes suivantes seront impactées par la nouvelle classe EditeurArticle :

Les trois méthodes suivantes seront déplacées dans la nouvelle classe :

- a. `string getEditeurName()`
- b. `string getEditeurCodePostal()`
- c. `string getEditeurAddress()`
- d. `void associerEditeur (string, string, string)`

Les méthodes suivantes devront être modifiées :

- e. `Article(string, string, string, string, string) → Article (string, string)`
- f. `void saveArticle(string)`
- g. `void afficher()`
- h. `Article* trouverArticle(string, string)`

La méthode suivante sera supprimée, car il y aura déjà une association (on ajoutera un objet de la classe EditeurArticle dans les attributs de la classe Article) :

- a. `void associerEditeur (string, string, string)`

4. Il faudra déplacer les attributs suivants dans la nouvelle classe EditeurArticle :
 - a. `string editeurName`
 - b. `string editeurCodePostal`
 - c. `string editeurAddress`

5. Le tableau ci-dessus énumère les différentes étapes à prendre pour restructurer cette odeur :

Étape	Description
Création de la nouvelle classe	On crée le fichier .h pour la classe EditeurArticle.
Écriture de la nouvelle classe	On met les définitions des méthodes et des attributs dans la nouvelle classe EditeurArticle comme indiquer aux points 3 et 4. On ajoute des setters pour chaque attributs.
Implémentation de la nouvelle classe	On crée le fichier .cpp de la classe EditeurArticle et on implémente ses fonctions.
Modification de la classe Article	On supprime la méthode indiquer au point 3. On modifie le constructeur par défaut. On ajoute un attribut de la classe EditeurArticle.
Modification de Article.cpp	On supprime les méthodes qui n'existent plus et on modifie les méthodes nécessaires.

6. Article.h :

```
#ifndef ARTICLE_H
#define ARTICLE_H

#include <string>
#include "EditeurArticle.h"

using namespace std;

// Cette classe représente un article
class Article {
public:
    // Constructeurs
    Article ();
    Article(string, string, string, string, string);

    // Setters
    void setName(string);
    void setAuteur(string);
    void associerEditeur (EditeurArticle*);

    // Getters
    string getName();
    string getAuteur();
    EditeurArticle* getEditeurArticle();

    // Enregistrer l'article
    void saveArticle(string);

    // Afficher l'article
    void afficher();

    // Chercher un article dans une base de données par nom
    Article* trouverArticle(string, string);

private:
    // Information sur l'article
    string name;
    string auteur;

    // Pour obtenir information sur l'editeur de l'article
    EditeurArticle* editeur;
};

#endif
```

Article.cpp :

```
#include "EditeurArticle.h"

#include <fstream>
#include <iostream>

EditeurArticle::EditeurArticle(){
    this->editeurName = "";
    this->editeurCodePostal = "";
    this->editeurAddress = "";
}

EditeurArticle::EditeurArticle(string editeurName, string editeurCodePostal, string editeurAdress){
    this->editeurName = editeurName;
    this->editeurCodePostal = editeurCodePostal;
    this->editeurAddress = editeurAdress;
}

//Setters
void EditeurArticle::setEditeurName(string editeurName){
    this->editeurName = editeurName;
}

void EditeurArticle::setEditeurCodePostal(string editeurCodePostal){
    this->editeurCodePostal = editeurCodePostal;
}

void EditeurArticle::setEditeurAddress(string editeurAddress){
    this->editeurAddress = editeurAddress;
}

//Getters
string EditeurArticle::getEditeurName(){
    return this->editeurName;
}

string EditeurArticle::getEditeurCodePostal(){
    return this->editeurCodePostal;
}

string EditeurArticle::getEditeurAddress(){
    return this->editeurAddress;
}
```

```

// Trouver un article avec son nom dans la base de données DB
Article* Article::trouverArticle (string DB, string name) {

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

    if(fichier){
        string line;
        // Lire les articles, un article par ligne dans la base de données (DB.txt)
        while (getline(fichier, line)) {
            string nameDB;
            // Récupérer le nom de l'article
            int i = 0;
            for (i = 0 ; i < line.length() ; i++) {
                if (line[i] != ',')
                    nameDB += line[i];
                else
                    break;
            }

            // Si l'article qu'on lit actuellement est celui qu'on cherche
            if (nameDB == name) {

                // Récupérer le nom de l'auteur
                string auteurDB;
                for (i = i + 1; i < line.length() ; i++) {
                    if (line[i] != ',')
                        auteurDB += line[i];
                    else
                        break;
                }

                // Récupérer le nom de l'éditeur
                string editeurNameDB;
                for (i = i + 1; i < line.length() ; i++) {
                    if (line[i] != ',')
                        editeurNameDB += line[i];
                    else
                        break;
                }

                // Récupérer le code postale de l'éditeur
                string editeurCodePostalDB;
                for (i = i + 1; i < line.length() ; i++) {
                    if (line[i] != ',')
                        editeurCodePostalDB += line[i];
                    else
                        break;
                }

                // Récupérer l'adresse de l'éditeur
                string editeurAddressDB;
                for (i = i + 1; i < line.length() ; i++) {
                    if (line[i] != ',')
                        editeurAddressDB += line[i];
                    else
                        break;
                }

                // Créer un objet de type article avec les informations récupérées
                Article *a = new Article(nameDB, auteurDB, editeurNameDB, editeurCodePostalDB, editeurAddressDB);
                // Fermer la base de données
                fichier.close();
                // Retourner l'article sélectionné
                return a;
            }
        }
        // Fermer la base de données
        fichier.close();
    }
    // Si l'article est inexistant, on retourne NULL
    return NULL;
}

```

```

// Afficher l'article
void Article::afficher(){
    std::cout << "Name : " << this->name << std::endl;
    std::cout << "Auteur : " << this->auteur << std::endl;
    std::cout << "Editeur name : " << (this->editeur)->getEditeurName() << std::endl;
    std::cout << "Editeur code postale : " << (this->editeur)->getEditeurCodePostal() << std::endl;
    std::cout << "Editeur address : " << (this->editeur)->getEditeurAddress() << std::endl;
}

```

EditeurArticle.h :

```

#ifndef EDITEURARTICLE_H
#define EDITEURARTICLE_H

#include <string>

using namespace std;
//cette classe groupe les informations de l'éditeur d'un article
class EditeurArticle{
public:
    //Constructeurs
    EditeurArticle();
    EditeurArticle(string, string, string);

    //Setters
    void setEditeurName(string);
    void setEditeurCodePostal(string);
    void setEditeurAddress(string);

    //Getters
    string getEditeurName();
    string getEditeurCodePostal();
    string getEditeurAddress();

private:
    // Informations sur l'éditeur de l'article
    string editeurName;
    string editeurCodePostal;
    string editeurAddress;
};
#endif

```

EditeurArticle.cpp :

```
#include "EditeurArticle.h"

#include <fstream>
#include <iostream>

EditeurArticle::EditeurArticle(){
    this->editeurName = "";
    this->editeurCodePostal = "";
    this->editeurAddress = "";
}

EditeurArticle::EditeurArticle(string editeurName, string editeurCodePostal, string editeurAdress){
    this->editeurName = editeurName;
    this->editeurCodePostal = editeurCodePostal;
    this->editeurAddress = editeurAdress;
}

//Setters
void EditeurArticle::setEditeurName(string editeurName){
    this->editeurName = editeurName;
}

void EditeurArticle::setEditeurCodePostal(string editeurCodePostal){
    this->editeurCodePostal = editeurCodePostal;
}

void EditeurArticle::setEditeurAddress(string editeurAddress){
    this->editeurAddress = editeurAddress;
}

//Getters
string EditeurArticle::getEditeurName(){
    return this->editeurName;
}

string EditeurArticle::getEditeurCodePostal(){
    return this->editeurCodePostal;
}

string EditeurArticle::getEditeurAddress(){
    return this->editeurAddress;
}
```


Compilation des tests :

```
[masare@l4712-15 tests (master)] $ ./ArticleTests
ArticleTest::testgetters : OK
ArticleTest::trouverArticle : OK
ArticleTest::trouverArticle2 : OK
OK (3)
[masare@l4712-15 tests (master)] $
```

Commit du code source :

```
[masare@l4712-15 log1000-28 (master)] $ git add .
[masare@l4712-15 log1000-28 (master)] $ git commit -m"Commit partie E1"
[master 5af89af] Commit partie E1
Committer: Mariam Sarwat <masare@l4712-15.info.polymtl.ca>
Votre nom et votre adresse courriel ont été configurés automatiquement en se
fondant sur votre nom d'utilisateur et le nom de votre machine. Veuillez
vérifier qu'ils sont corrects. Vous pouvez supprimer ce message en les
paramétrant explicitement. Lancez les commandes suivantes et suivez les
instruction dans votre éditeur pour éditer votre fichier de configuration :

    git config --global --edit

Après ceci, vous pouvez corriger l'identité utilisée pour cette validation avec :

    git commit --amend --reset-author

20 files changed, 26 insertions(+), 55 deletions(-)
rename TP5/{TP5 => }/LOG1000_TP5.pdf (100%)
delete mode 100644 TP5/TP5/.DS_Store
delete mode 100644 TP5/TP5/sources/Article.o
delete mode 100644 TP5/TP5/sources/EditeurArticle.o
delete mode 100755 TP5/TP5/sources/article
delete mode 100644 TP5/TP5/sources/main.o
delete mode 100755 TP5/TP5/sources/tests/ArticleTests
rename TP5/{TP5 => }/sources/.DS_Store (100%)
rename TP5/{TP5 => }/sources/Article.cpp (79%)
rename TP5/{TP5 => }/sources/Article.h (100%)
rename TP5/{TP5 => }/sources/DB.txt (100%)
rename TP5/{TP5 => }/sources/EditeurArticle.cpp (95%)
rename TP5/{TP5 => }/sources/EditeurArticle.h (100%)
rename TP5/{TP5 => }/sources/Makefile (96%)
rename TP5/{TP5 => }/sources/main.cpp (100%)
rename TP5/{TP5 => }/sources/tests/ArticleTest.h (80%)
create mode 100755 TP5/sources/tests/ArticleTests
rename TP5/{TP5 => }/sources/tests/Makefile (100%)
rename TP5/{TP5 => }/sources/tests/main.cpp (100%)
rename TP5/{TP5 => }/tp5_log1000.docx (100%)
[masare@l4712-15 log1000-28 (master)] $ git push
Username for 'https://github.com': masare
Password for 'https://masare@github.com':
Décompte des objets: 10, fait.
Delta compression using up to 8 threads.
Compression des objets: 100% (10/10), fait.
Écriture des objets: 100% (10/10), 95.99 KiB | 2.00 MiB/s, fait.
Total 10 (delta 1), reused 0 (delta 0)
To https://github.com:masare/log1000-28
 dc91680..5af89af master -> master
[masare@l4712-15 log1000-28 (master)] $
```

E2) Une mauvaise odeur dans les méthodes

1. L'odeur grave qui se trouve dans la méthode « TrouverArticle » de la classe « Article.cpp » est une de code dupliqué. En effet, pour récupérer le nom de l'auteur, le nom de l'éditeur, son code postal et son adresse, on utilise la même boucle d'itération. Ceci est une odeur grave, puisque la méthode est devenue répétitive et lourde lorsqu'elle pourrait être beaucoup plus courte.

2.

Étape	Description
Déclaration et implémentation d'une nouvelle méthode dans la classe Article : lireArticle()	Au lieu d'avoir une grosse méthode (TrouverArticle), on crée une nouvelle méthode lireArticle qui ira lire les articles à la ligne indiquer.
Déclaration et implémentation d'une nouvelle méthode dans la classe Article : recupererArticle()	Au lieu d'avoir une grosse méthode (TrouverArticle), on crée une nouvelle méthode recupererArticle qui récupèrera l'information de l'article une fois trouver (à l'aide de la fonction TrouverArticle).

3.

Nouvelle méthode string Article :: recupererArticle(string ligne, int& positionLecture) :

```
// Trouver un article avec son nom dans la base de données DB
string Article::recupererArticle(string ligne, int& positionLecture){
    string variable; //peut etre nom, editeur, codepostal ou adresse
    while(positionLecture<ligne.length()){
        if(ligne[positionLecture] !=','){
            variable+=ligne[positionLecture];
        }
        else
            break;
        positionLecture++;
    }

    ++positionLecture;
    return variable;
}
```

Modification de la méthode Article* Article :: trouverArticle(string DB, string name) :

```
Article* Article::trouverArticle (string DB, string name) {  
  
    ifstream fichier(DB.c_str(), ios::in); // Ouvrir le fichier "DB.txt"  
  
    if(fichier) {  
        string line;  
        // Lire les articles, un article par ligne dans la base de données (DB.txt)  
        while (getline(fichier, line)) {  
            int i = 0;  
            // Récupérer le nom de l'article  
            string nameDB = recupererArticle(line, i);  
  
            // Si l'article qu'on lit actuellement est celui qu'on cherche  
            if (nameDB == name) {  
  
                // Récupérer le nom de l'auteur  
                string auteurDB = recupererArticle(line, i);  
  
                // Récupérer le nom de l'éditeur  
                string editeurNameDB = recupererArticle(line, i);  
  
                // Récupérer le code postale de l'éditeur  
                string editeurCodePostalDB = recupererArticle(line, i);  
  
                // Récupérer l'adresse de l'éditeur  
                string editeurAddressDB = recupererArticle(line, i);  
  
                // Créer un objet de type article avec les informations récupérées  
                Article *a = new Article(nameDB, auteurDB, editeurNameDB, editeurCodePostalDB, editeurAddressDB);  
                // Fermer la base de données  
                fichier.close();  
                // Retourner l'article sélectionné  
                return a;  
            }  
        }  
        // Fermer la base de données  
        fichier.close();  
    }  
    // Si l'article est inexistant, on retourne NULL  
    return NULL;  
}
```

Compilation des tests :

```
[masare@l4712-15 tests (master)] $ ./ArticleTests  
ArticleTest::testgetters : OK  
ArticleTest::trouverArticle : OK  
ArticleTest::trouverArticle2 : OK  
OK (3)  
[masare@l4712-15 tests (master)] $
```

Commit du code source :

```
[masare@l4712-15 log1000-28 (master)] $ git add .
[masare@l4712-15 log1000-28 (master)] $ git commit -m"Ajout partie 2"
[master 752a751] Ajout partie 2
  Committer: Mariam Sarwat <masare@l4712-15.info.polymtl.ca>
Votre nom et votre adresse courriel ont été configurés automatiquement en se
fondant sur votre nom d'utilisateur et le nom de votre machine. Veuillez
vérifier qu'ils sont corrects. Vous pouvez supprimer ce message en les
paramétrant explicitement. Lancez les commandes suivantes et suivez les
instruction dans votre éditeur pour éditer votre fichier de configuration :
```

```
git config --global --edit
```

Après ceci, vous pouvez corriger l'identité utilisée pour cette validation avec :

```
git commit --amend --reset-author
```

```
6 files changed, 36 insertions(+), 42 deletions(-)
create mode 100644 .gitignore
rewrite TP5/sources/tests/ArticleTests (76%)
rewrite TP5/tp5_log1000.docx (65%)
[masare@l4712-15 log1000-28 (master)] $ git push
Username for 'https://github.com': masare
Password for 'https://masare@github.com':
Décompte des objets: 11, fait.
Delta compression using up to 8 threads.
Compression des objets: 100% (10/10), fait.
Écriture des objets: 100% (11/11), 650.40 KiB | 21.68 MiB/s, fait.
Total 11 (delta 7), reused 0 (delta 0)
To https://github.com/mariam-sarwat/log1000-28
  5af89af..752a751 master -> master
[masare@l4712-15 log1000-28 (master)] $
```

E3) Utilisation des variables

1.

DBFile :

- a. $\text{Span moyen} = (34+9)/2=21,5$
- b. $\text{Durée de vie} = (52-7)=45$
- c. $\text{Portée} = (60-5)=55$

Opération :

- a. $\text{Span moyen} = (8+0+37)/3=15$
- b. $\text{Durée de vie} = (56-8)=48$
- c. $\text{Portée} = (60-5)=55$

Article :

- a. $\text{Span moyen} = (28+0+5+0+3+0+3+0)/8=4,88$
- b. $\text{Durée de vie} = (53-6)=47$
- c. Portée :
 - i. $(35-5)=30$
 - ii. $(53-36)=17$

2. La variable qui bénéficiera le plus d'une restructuration est DBFile, car elle a le span moyen le plus élevé.

3.

Étape	Description
Rapprocher la déclaration de la variable	On rapproche la déclaration de la variable pour diminuer sa portée (et ainsi diminuer son span moyen)
Rapprocher l'utilisation de la variable	On rapproche l'utilisation de la variable pour diminuer sa portée lorsque possible (et ainsi diminuer son span moyen)

4. main.cpp :

```
#include <iostream>
#include "Article.h"

using namespace std;

/*
 * fonction principale
 */
int main(int argc, char** argv) {

    Article* article = new Article(); // Création d'un article

    int operation ; // Opération sélectionnée par l'utilisateur

    do {
        // Afficher les opérations possibles
        std::cout << std::endl << "-----" << std::endl;

        std::cout << "0 - Quitter le programme " << std::endl;
        std::cout << "1 - Créer un article " << std::endl;
        std::cout << "2 - Trouver un article " << std::endl;
        std::cout << "3 - Afficher un article " << std::endl;
        std::cout << "4 - Enregistrer l'article " << std::endl;
        std::cout << "-----" << std::endl;

        // Lire le choix d'utilisateur
        std::cin >> operation;

        string DBFile = "DB.txt"; // Fichier qui contient une base des articles

        switch (operation) {
            case 1:
            {
                // Informations du nouvel article
                string name;
                string auteur;
                string editeurName;
                string editeurCodePostal;
                string editeurAddress;

                // Demander l'utilisateur de saisir les informations du nouvel article
                std::cout << "Saisir le nom de l'article : " ;
                std::cin >> name;

                std::cout << "Saisir l'auteur de l'article : " ;
                std::cin >> auteur;

                std::cout << "Saisir le nom de l'editeur : " ;
                std::cin >> editeurName;

                std::cout << "Saisir le code postale de l'editeur : " ;
                std::cin >> editeurCodePostal;

                std::cout << "Saisir l'adresse de l'editeur : " ;
                std::cin >> editeurAddress;

                // Créer un nouvel article
                delete article;

                article = new Article (name, auteur, editeurName, editeurCodePostal, editeurAddress);

                break;
            }
        }
    } while (operation != 0);
}
```

```

case 2:
{
    // Demander l'utilisateur de saisir le nom d'article à chercher dans la base de données
    string name;
    std::cout << "Saisir le nom de l'article : " ;
    std::cin >> name;

    // Chercher l'article
    Article* tmp = article->trouverArticle(DBFile, name);

    if (tmp != NULL) {
        // Si l'article est trouvé
        article = tmp;
    } else {
        std::cout << "Aucun article trouvé !" << std::endl;
    }

    break;
}
case 3:
{
    // Afficher l'article
    if (article != NULL) {
        article->afficher();
    } else {
        std::cout << "Aucun article sélectionné" << std::endl;
    }
    break;
}
case 4:
{
    // Enregistrer l'article dans la base de données.
    if (article != NULL) {
        article->saveArticle(DBFile);
        std::cout << "Article enregistré !" << std::endl;
    }
}
}

} while (operation != 0); // Tant que l'utilisateur ne décide pas de quitter le programme

return 0;
}

```

Compilation et test des opérations :

```
(masare@l4712-15 sources (master)) $ ./article

-----
0 - Quitter le programme
1 - Créer un article
2 - Trouver un article
3 - Afficher un article
4 - Enregistrer l'article
-----
2
Saisir le nom de l'article : 0
Aucun article trouvé !

-----
0 - Quitter le programme
1 - Créer un article
2 - Trouver un article
3 - Afficher un article
4 - Enregistrer l'article
-----
1
Saisir le nom de l'article : log1000
Saisir l'auteur de l'article : steph
Saisir le nom de l'editeur : mariam
Saisir le code postale de l'editeur : h3j6j0
Saisir l'adresse de l'editeur : 456

-----
0 - Quitter le programme
1 - Créer un article
2 - Trouver un article
3 - Afficher un article
4 - Enregistrer l'article
-----
2
Saisir le nom de l'article : infl600
Aucun article trouvé !

-----
0 - Quitter le programme
1 - Créer un article
2 - Trouver un article
3 - Afficher un article
4 - Enregistrer l'article
-----
3
Name : log1000
Auteur : steph
Editeur name : mariam
Editeur code postale : h3j6j0
Editeur address : 456

-----
0 - Quitter le programme
1 - Créer un article
2 - Trouver un article
3 - Afficher un article
4 - Enregistrer l'article
-----
4
Article enregistré !

-----
0 - Quitter le programme
1 - Créer un article
2 - Trouver un article
3 - Afficher un article
4 - Enregistrer l'article
-----
0
```


Commit du code source :

```
[masare@l4712-15 log1000-28 (master)] $ git add .
[masare@l4712-15 log1000-28 (master)] $ git commit -m"Ajout partie e3"
[master 3ec31df] Ajout partie e3
  Committer: Mariam Sarwat <masare@l4712-15.info.polymtl.ca>
Votre nom et votre adresse courriel ont été configurés automatiquement en se
fondant sur votre nom d'utilisateur et le nom de votre machine. Veuillez
vérifier qu'ils sont corrects. Vous pouvez supprimer ce message en les
paramétrant explicitement. Lancez les commandes suivantes et suivez les
instruction dans votre éditeur pour éditer votre fichier de configuration :
```

```
git config --global --edit
```

Après ceci, vous pouvez corriger l'identité utilisée pour cette validation avec :

```
git commit --amend --reset-author
```

```
4 files changed, 95 insertions(+), 8 deletions(-)
delete mode 100644 TP5/sources/.DS_Store
create mode 100644 TP5/sources/Nouveau document
[masare@l4712-15 log1000-28 (master)] $ git push
Username for 'https://github.com': masare
Password for 'https://masare@github.com':
Décompte des objets: 7, fait.
Delta compression using up to 8 threads.
Compression des objets: 100% (7/7), fait.
Écriture des objets: 100% (7/7), 89.27 KiB | 3.31 MiB/s, fait.
Total 7 (delta 4), reused 0 (delta 0)
To https://github.com:masare/log1000-28
  752a751..3ec31df  master -> master
[masare@l4712-15 log1000-28 (master)] $
```