

École Polytechnique de Montréal  
Département de génie informatique et génie logiciel

LOG1000  
Ingénierie logicielle

TP5  
Restructuration du code source

Soumis par :  
William Harvey (1851388) et  
Mathieu Bélanger (1850591)

19 avril 2017



## E1) Une mauvaise odeur dans les attributs

1. Il s'agit d'une odeur grave, car il s'agit d'une trop grosse classe. En effet, cette classe possède beaucoup trop d'attributs, puisqu'elle englobe deux abstractions différentes. La première est la représentation d'une émission de télévision et la deuxième est la représentation d'une chaîne de télévision. Ainsi, on a une classe qui devrait être en fait représentée par deux classes.
2. Le nom de la restructuration nécessaire pour enlever cette odeur du code est l'extraction d'une classe. En effet, pour résoudre cette mauvaise odeur, il suffit de créer une nouvelle classe Chaîne et de lui donner les attributs et les méthodes liées à la chaîne de télévision. La classe Emission ne conserve alors que les attributs et les méthodes propres au concept d'une émission de télévision. Ensuite, on peut établir, par exemple, une relation d'agrégation entre les classe Chaîne et Emission, puisqu'un chaîne de télévision diffuse des émissions.
3. Les méthodes qui sont impactées par la nouvelle classe Chaîne sont:
  - a) Emission(string, string, string, string, string); → Emission(string, string);
  - b) void associerChaîne(string, string, string);
  - c) string getChaîneName(); → Déplacement vers la nouvelle classe Chaîne
  - d) string getChaîneCodePostal(); → Déplacement vers la nouvelle classe Chaîne
  - e) string getChaîneAddress(); → Déplacement vers la nouvelle classe Chaîne
4. Les attributs qu'il faut déplacer sont:
  - a. string chaîneName;
  - b. string chaîneCodePostal;
  - c. string chaîneAddress;

5.

Étape	Description
Création de la nouvelle classe Chaîne	Création d'un fichier .cpp et .h pour la classe Chaîne.
Écriture de Chaîne.h	Ajout des prototypes et des attributs de la classe Chaîne. Ajout des setters pour les attributs propre à la classe Chaîne
Écriture de Chaîne.cpp	Ajout des définitions des méthodes de la classe Chaîne
Modification Emission.h	Suppression des prototypes et des attributs, maintenant dans Chaîne.h, de la classe Emission
Modification Emission.cpp	Suppression des définitions de méthode désormais inexistantes dans Emission.h

6.

## Emission.h

```
#ifndef Emission_H
#define Emission_H

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

using namespace std;

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

    // Setters
    void setTitre(string);
    void setAnimateur(string);
    void associerChaine(Chaine*);

    // Getters
    string getTitre();
    string getAnimateur();
    Chaine* getChaine();

    // Enregistrer l'Emission
    void saveEmission(string);
    // Afficher l'Emission
    void afficher();
    // Chercher un Emission dans une base de données par titre
    Emission* trouverEmission(string, string);

private:
    // Information sur l'emission
    string titre;
    string animateur;
    Chaine* chaine;
};

#endif
```

## Emission.cpp

```
#include "Emission.h"
#include <fstream>
#include <iostream>

// Constructeur
Emission::Emission(){
    titre = "";
    animateur = "";
    chaine = new Chaine();
}

Emission::Emission (string titre,
                    string animateur,
                    string chaineName,
```

```

        string chaineCodePostal,
        string chaineAddress) {
    // Emission information
    this->titre = titre;
    this->animateur = animateur;
    this->chaine = new Chaine (chaineName, chaineCodePostal, chaineAddress);
}

// Setters
void Emission::setTitre(string titre) {
    this->titre = titre;
}
void Emission::setAnimateur(string animateur) {
    this->animateur = animateur;
}

// Associer une Chaine à l'Emission
void Emission::associerChaine (Chaine* chaine) {
    this -> chaine = chaine;
}

// Getters
string Emission::getTitre() {
    return this->titre;
}
string Emission::getAnimateur() {
    return this->animateur;
}
Chaine* Emission::getChaine() {
    return this->chaine;
}

// Enregistrer l'Emission dans un fichier
void Emission::saveEmission (string fileName) {

    ofstream outfile (fileName.c_str(), std::ofstream::binary | std::fstream::app);
    // write to outfile
    outfile<<this->titre <<","
            <<this->animateur <<"\n";
    outfile.close();
}

// 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)) {
            string titreDB;
            // Récupérer le nom de l'Emission
            int i = 0;
            for (i = 0 ; i < line.length() ; i++) {
                if (line[i] != ',') {
                    titreDB += line[i];
                } else {
                    break;
                }
            }
        }
    }
}

```

```

    }
}

// Si l'Emission qu'on lit actuellement est celui qu'on cherche
if (titreDB == titre) {

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

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

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

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

    // 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;

```

```

}

// Afficher l'Emission
void Emission::afficher() {
    std::cout << "Titre : " << this->titre << std::endl;
    std::cout << "Animateur : " << this->animateur << std::endl;
    std::cout << "Chaine name : " << (this->chaine)->getChaineName() << std::endl;
    std::cout << "Chaine code postale : " << (this->chaine)->getChaineCodePostal() <<
std::endl;
    std::cout << "Chaine adresse : " << (this->chaine)->getChaineAddress() << std::endl;
}

```

## Chaine.h

```

#ifndef Chaine_H
#define Chaine_H

#include <string>

using namespace std;

// Cette classe représente une Chaine
class Chaine {
public:
    // Constructeurs
    Chaine ();
    Chaine(string, string, string);

    // Setters

    void setChaineAdress(string);
    void setName(string);
    void setCodePostal(string);

    // Getters
    string getChaineName();
    string getChaineCodePostal();
    string getChaineAddress();

    //Affichage
    void afficher();

private:
    // Informations sur la chaine
    string chaineName;
    string chaineCodePostal;
    string chaineAddress;
};

#endif

```

## Chaine.cpp

```

#include "Chaine.h"

```

```

#include <fstream>
#include <iostream>

// Constructeurs
Chaine::Chaine(){
    this->chaineName = "";
    this->chaineCodePostal = "";
    this->chaineAddress = "";
}

Chaine::Chaine (string chaineName,
                string chaineCodePostal,
                string chaineAddress) {

    // Chaine information
    this->chaineName = chaineName;
    this->chaineCodePostal = chaineCodePostal;
    this->chaineAddress = chaineAddress;
}

// Setters
void Chaine::setName(string chaineName) {
    this->chaineName = chaineName;
}

void Chaine::setCodePostal(string chaineCodePostal) {
    this->chaineCodePostal = chaineCodePostal;
}

void Chaine::setChaineAddress(string chaineAddress) {
    this->chaineAddress = chaineAddress;
}

// Getters

string Chaine::getChaineName() {
    return this->chaineName;
}

string Chaine::getChaineCodePostal() {
    return this->chaineCodePostal;
}

string Chaine::getChaineAddress() {
    return this->chaineAddress;
}

// Afficher la chaine
void Chaine::afficher() {
    std::cout << "Chaine name : " << this->chaineName << std::endl;
    std::cout << "Chaine code postale : " << this->chaineCodePostal << std::endl;
    std::cout << "Chaine address : " << this->chaineAddress << std::endl;
}

```

```
[wihar@l4714-15 test]$ ./EmissionTests
EmissionTest::testgetters : OK
EmissionTest::trouverEmission : OK
EmissionTest::trouverEmission2 : OK
OK (3)
[wihar@l4714-15 test]$
```

8.



```
[wihar@l4714-15 log1000-62]$ git pull
Username for 'https://github.com': wihaar
Password for 'https://wihaar@github.com':
Already up-to-date.
[wihar@l4714-15 log1000-62]$ git add --all
[wihar@l4714-15 log1000-62]$ git commit -m "Ajout E1) TP5"
[master 165d867] Ajout E1) TP5
Committer: William Harvey <wihaar@l4714-15.info.polymtl.ca>
Votre nom et votre adresse e-mail 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
```

```
11 files changed, 36 insertions(+), 21 deletions(-)
create mode 100644 tp5/sources/Chaine.o
create mode 100755 tp5/sources/Emission
create mode 100644 tp5/sources/Emission.o
create mode 100644 tp5/sources/main.o
create mode 100755 tp5/sources/test/EmissionTests
[wihar@l4714-15 log1000-62]$ git push
warning: push.default n'est pas défini ; sa valeur implicite a changé dans Git 2.0
de 'matching' vers 'simple'. Pour supprimer ce message et maintenir
le comportement actuel après la modification de la valeur de défaut, utilisez :
```

```
git config --global push.default matching
```

Pour supprimer ce message et adopter le nouveau comportement maintenant, utilisez :

```
git config --global push.default simple
```

Quand push.default vaudra 'matching', git poussera les branches locales sur les branches distantes qui existent déjà avec le même nom.

Depuis Git 2.0, Git utilise par défaut le comportement plus conservatif 'simple' qui ne pousse la branche courante que vers la branche distante correspondante que 'git pull' utilise pour mettre à jour la branche courante.

Voir 'git help config' et chercher 'push.default' pour plus d'information. (le mode 'simple' a été introduit dans Git 1.7.11. Utilisez le mode similaire 'current' au lieu de 'simple' si vous utilisez de temps en temps d'anciennes versions de Git)

```
Username for 'https://github.com': wihaar
Password for 'https://wihaar@github.com':
Décompte des objets: 16, fait.
Delta compression using up to 4 threads.
Compression des objets: 100% (16/16), fait.
Écriture des objets: 100% (16/16), 69.77 KiB | 0 bytes/s, fait.
Total 16 (delta 7), reused 0 (delta 0)
To https://github.com:polymtl-ca/git/log1000-62
 9f8d29d..165d867 master -> master
[wihar@l4714-15 log1000-62]$ git pull
Username for 'https://github.com': wihaar
Password for 'https://wihaar@github.com':
Already up-to-date.
[wihar@l4714-15 log1000-62]$
```

## E2) Une mauvaise odeur dans les méthodes

### 1.

- a) Code mort. La variable pointeur *tmp* déclarée en début de méthode n'est jamais utilisée par la suite. *Emission\*tmp=NULL;*
- b) Code dupliqué. La section d'itération sur les lignes (for loop) pour récupérer le titre, l'animateur, la chaine, etc. est dupliquée 5 fois.

2.

Étape	Description
Suppression de la variable morte	La variable est inutilisée. On peut donc retirer sa déclaration
déclaration d'une méthode Emission::lireEmissionDB() dans Emission.h, puis implémentation dans Emission.cpp	La méthode sera appelée après que l'émission voulue (titreDB == titre) ait été trouvée
déclaration d'une méthode Emission::lireProchainElem() dans Emission.h, puis implémentation dans Emission.cpp	La méthode est appelé à chaque fois que l'on souhaite lire le prochain élément (jusqu'à la prochaine virgule) d'une string

3.

```
string Emission::lireProchainElem(string line, int &indiceCourant) {
    string tmp = "";
    for (indiceCourant = indiceCourant + 1; indiceCourant < line.length() ; indiceCourant++) {
        if (line[indiceCourant] != ',') {
            tmp += line[indiceCourant];
        }
        else {
            return tmp;
        }
    }
}
```

```
Emission* Emission::lireEmissionDB(string titreDB, string line, int indiceCourant) {
```

```

// Récupérer le nom de l'animateur
string animateurDB = lireProchainElem(line, indiceCourant);

// Récupérer le nom de l'éditeur
string chaineNameDB = lireProchainElem(line, indiceCourant);

// Récupérer le code postale de l'éditeur
string chaineCodePostalDB = lireProchainElem(line, indiceCourant);

// Récupérer l'adresse de l'éditeur
string chaineAddressDB = lireProchainElem(line, indiceCourant);

return new Emission(titreDB, animateurDB, chaineNameDB, chaineCodePostalDB,
chaineAddressDB);
}

```

```

// 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)) {
            string titreDB;
            // Récupérer le nom de l'Emission
            int iterateurDeLigne = -1;
            titreDB = lireProchainElem(line,iterateurDeLigne);

            // Si l'Emission qu'on lit actuellement est celui qu'on cherche
            if (titreDB == titre) {

                // Fermer la base de données
                fichier.close();
                // Retourner l'Emission sélectionné
                return lireEmissionDB(titreDB, line, iterateurDeLigne);
            }
        }
        // Fermer la base de données
        fichier.close();
    }
    // Si l'Emission est innexistant, on retourne NULL
    return NULL;
}

```

4.

```
[mabelam@l4714-04 test]$ make
g++ -g -I./ -o EmissionTests main.cpp ../Emission.o -lcppunit
[mabelam@l4714-04 test]$ ./EmissionTests
EmissionTest::testgetters : OK
EmissionTest::trouverEmission : OK
EmissionTest::trouverEmission2 : OK
OK (3)
[mabelam@l4714-04 test]$
```

5.

```
[mabelam@l4714-04 log1000-62]$ git pull
Username for 'https://github.com': mabelam
Password for 'https://mabelam@github.com':
Already up-to-date.
[mabelam@l4714-04 log1000-62]$ git add tp5
[mabelam@l4714-04 log1000-62]$ git commit -m "ajout E2"
[master 0933dce] ajout E2
 5 files changed, 43 insertions(+), 55 deletions(-)
 rewrite tp5/sources/Emission (89%)
 rewrite tp5/sources/Emission.o (78%)
[mabelam@l4714-04 log1000-62]$ git status
Sur la branche master
Votre branche est en avance sur 'origin/master' de 1 commit.
(utilisez "git push" pour publier vos commits locaux)
Modifications qui ne seront pas validées :
  (utilisez "git add/rm <fichier>..." pour mettre à jour ce qui sera validé)
  (utilisez "git checkout -- <fichier>..." pour annuler les modifications dans la copie de travail)

    supprimé :      TP1/data/.DS_Store
    supprimé :      TP1/src/.DS_Store
    supprimé :      TP4/tests/.DS_Store

aucune modification n'a été ajoutée à la validation (utilisez "git add" ou "git commit -a")
[mabelam@l4714-04 log1000-62]$ git push
warning: push.default n'est pas défini ; sa valeur implicite a changé dans Git 2.0
de 'matching' vers 'simple'. Pour supprimer ce message et maintenir
le comportement actuel après la modification de la valeur de défaut, utilisez :

    git config --global push.default matching

Pour supprimer ce message et adopter le nouveau comportement maintenant, utilisez :

    git config --global push.default simple

Quand push.default vaudra 'matching', git poussera les branches locales
sur les branches distantes qui existent déjà avec le même nom.

Depuis Git 2.0, Git utilise par défaut le comportement plus conservatif 'simple'
qui ne pousse la branche courante que vers la branche distante correspondante
que 'git pull' utilise pour mettre à jour la branche courante.

Voir 'git help config' et chercher 'push.default' pour plus d'information.
(le mode 'simple' a été introduit dans Git 1.7.11. Utilisez le mode similaire
'current' au lieu de 'simple' si vous utilisez de temps en temps d'anciennes versions de Git)

Username for 'https://github.com': mabelam
Password for 'https://mabelam@github.com':
Décompte des objets: 9, fait.
Delta compression using up to 4 threads.
Compression des objets: 100% (9/9), fait.
Écriture des objets: 100% (9/9), 7.94 KiB | 0 bytes/s, fait.
Total 9 (delta 7), reused 0 (delta 0)
To https://github.com:mabelam/log1000-62
 165d867..0933dce master -> master
[mabelam@l4714-04 log1000-62]$ gitk
```

### E3) Utilisation des variables

#### 1. DBFile:

- a.  $\text{Span moyen} = (41+16)/2 = 28,5$
- b. Durée de vie = 61
- c. Portée = 64

#### choix:

- a.  $\text{Span moyen} = (10 + 0 + 47) / 3 = 19$
- b. Durée de vie = 61
- c.  $\text{Portée} = 85 - 9 = 62$

#### emission:

- a.  $\text{Span moyen} = (33 + 0 + 7 + 1 + 6 + 0 + 5 + 0) / 8 = 6,5$
- b. Durée de vie = 61
- c. Portée = 64

2. La variable qui bénéficiera le plus de la restructuration est celle qui a le span moyen le plus élevé. En effet, c'est le paramètre le plus décisive en ce qui a trait à la clarté. Ainsi, c'est la variable DBFile qui a le plus besoin d'une restructuration.

3.

Étape	Description
Réduction de la portée de la variable	En rapprochant la déclaration de la variable de sa première utilisation, on réduit la portée de cette dernière ce qui permet aussi de réduire le span moyen.
Rapprocher l'utilisation de la variable de sa déclaration.	En déplaçant les cas 3 et 4 aux cas 1 et 2 et vis-versa. On rapproche l'utilisation de la variable de sa déclaration. Ainsi, on réduit le span moyen.

4.

```
#include <iostream>
#include "Emission.h"
#include "Chaine.h"

using namespace std;

/*
 * fonction principale
 */
int main(int argc, char** argv) {
    Emission* emission = new Emission(); // Création d'un emission
    int choix; // 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 - Trouver une emission " << std::endl;
std::cout << "2 - Enregistrer l'émission " << std::endl;
std::cout << "3 - Créer une emission " << std::endl;
std::cout << "4 - Afficher une emission " << std::endl;
std::cout << "-----" << std::endl;
// Lire le choix d'utilisateur
std::cin >> choix;

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

switch (choix) {
    case 1:
    {
        // Demander l'utilisateur de saisir le nom d'emission à chercher dans la base
de données
        string titre;
        std::cout << "Saisir le titre de titre l'émission : " ;
        std::cin >> titre;
        // Chercher l'emission
        Emission* tmp = emission->trouverEmission(DBFile, titre);
        if (tmp != NULL) { // Si l'emission est trouvé
            emission = tmp;
            std::cout << "Emission trouvée !" << std::endl;
        } else {
            std::cout << "Aucune émission trouvée !" << std::endl;
        }
        break;
    }
    case 2:
    {
        // Enregistrer l'emission dans la base de données.
        if (emission != NULL) {
            emission->saveEmission(DBFile);
            std::cout << "Emission enregistrée !" << std::endl;
        }
        break;
    }
    case 3:
    {
        // Informations du nouvel emission
        string titre;
        string animateur;
        string chaineName;
        string chaineCodePostal;
        string chaineAddress;
        // Demander l'utilisateur de saisir les informatins du nouvel emission
        std::cout << "Saisir le titre de l'émission : " ;
        std::cin >> titre;
        std::cout << "Saisir l'animateur de l'émission : " ;
        std::cin >> animateur;
        std::cout << "Saisir le nom de la chaine : " ;
        std::cin >> chaineName;
        std::cout << "Saisir le code postale de la chaine : " ;
        std::cin >> chaineCodePostal;
        std::cout << "Saisir l'adresse de la chaine : " ;
        std::cin >> chaineAddress;
        // Créer un nouvel emission
        delete emission;
        emission = new Emission (titre, animateur, chaineName, chaineCodePostal,

```

```

chaineAddress);
    break;
}

    case 4:
    {
        // Afficher l'emission
        if (emission != NULL) {
            emission->afficher();
        } else {
            std::cout << "Aucune émission sélectionnée" << std::endl;
        }
        break;
    }

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

```

5.

```
++ -o Emission main.o Emission.o Chaine.o
vihar@l4714-15 sources]$ ./Emission

-----
- Quitter le programme
- Trouver une emission
- Enregistrer l'émission
- Créer une emission
- Afficher une emission
-----

Saisir le titre de l'émission : titreTest1
Emission trouvée !

-----
- Quitter le programme
- Trouver une emission
- Enregistrer l'émission
- Créer une emission
- Afficher une emission
-----

Emission enregistrée !

-----
- Quitter le programme
- Trouver une emission
- Enregistrer l'émission
- Créer une emission
- Afficher une emission
-----

Saisir le titre de l'émission : titreTest8
Saisir l'animateur de l'émission : EricSalvaille
Saisir le nom de la chaine : yoLesJeunesTV
Saisir le code postale de la chaine : XXXXXX
Saisir l'adresse de la chaine : 123rueDeLaMontagne

-----
- Quitter le programme
- Trouver une emission
- Enregistrer l'émission
- Créer une emission
- Afficher une emission
-----

Titre : titreTest8
Animateur : EricSalvaille
Chaine name : yoLesJeunesTV
Chaine code postale : XXXXXX
Chaine adresse : 123rueDeLaMontagne

-----
- Quitter le programme
- Trouver une emission
- Enregistrer l'émission
- Créer une emission
- Afficher une emission
-----

vihar@l4714-15 sources]$
```



6.

```
g++ -o Emission main.o Emission.o Chaine.o
[wihar@l4714-15 sources]$ ./Emission

-----
0 - Quitter le programme
1 - Trouver une emission
2 - Enregistrer l'émission
3 - Créer une emission
4 - Afficher une emission
-----
1
Saisir le titre de titre l'émission : titreTest1
Emission trouvée !

-----
0 - Quitter le programme
1 - Trouver une emission
2 - Enregistrer l'émission
3 - Créer une emission
4 - Afficher une emission
-----
2
Emission enregistrée !

-----
0 - Quitter le programme
1 - Trouver une emission
2 - Enregistrer l'émission
3 - Créer une emission
4 - Afficher une emission
-----
3
Saisir le titre de l'émission : titreTest8
Saisir l'animateur de l'émission : EricSalvaille
Saisir le nom de la chaine : yoLesJeunesTV
Saisir le code postale de la chaine : XXXXXX
Saisir l'adresse de la chaine : 123rueDeLaMontagne

-----
0 - Quitter le programme
1 - Trouver une emission
2 - Enregistrer l'émission
3 - Créer une emission
4 - Afficher une emission
-----
4
Titre : titreTest8
Animateur : EricSalvaille
Chaine name : yoLesJeunesTV
Chaine code postale : XXXXXX
Chaine adresse : 123rueDeLaMontagne

-----
0 - Quitter le programme
1 - Trouver une emission
2 - Enregistrer l'émission
3 - Créer une emission
4 - Afficher une emission
-----
0
[wihar@l4714-15 sources]$
```

## Bonus

3 Avantages	3 Inconvénients
Apprentissage de l'utilisation de logiciel sous licence open source. Éduque quant à la possibilité de réutiliser du code source.	Le fait que notre travail ne soit pas réellement intégré au sein des projets dévalorise un peu l'effort.
Apprentissage de diverses plateformes de travail collaboratif tel que Git ou encore Gerrit	La complexité des logiciels open source utilisés peut être déroutante et ce, même dans un cadre éducatif. Ainsi, il était parfois difficile de compléter certaines tâches sans faire une tentative approximative. (Ex. schéma de classe de Ring)
L'utilisation de projets open source permet de montrer un cas réel de développement logiciel et toutes ses problématiques. Il aurait été impossible de démontrer des cas de problématique de développement logiciel avec du code fermé.	Malgré le fait que nous avons utilisé des projets open-source, l'aspect collaboratif de ces projets, à notre sens, n'a pas été assez abordé. En effet, nous travaillons en équipe isolée, afin de contribuer à un projet commun.