

École de technologie supérieure	Trimestre	Hiver 2016
Département de génie logiciel et des TI	Professeur	Stéphane Coulombe
	Préalable	LOG120
	Crédits	4

GTI310 Structure de données multimédias

Deuxième Travail Pratique

Sujet : Manipulation de fichiers audio

Description

La compagnie Audio-Di-Dac veut être en mesure d'éditer les fichiers de son qu'elle enregistre. Elle aimerait avoir un programme en mesure de modifier un paramètre de façon très particulière pour l'ensemble des fichiers de son dont elle dispose.

Votre mandat consiste à développer un filtre audio pour convertir des valeurs de 16 bits, à des valeurs de 8 bits. Le programme doit analyser un fichier « .wav » non compressé, transformer la valeur de 16 bits en une valeur de 8 bits et produire un nouveau fichier « .wav » à la sortie. Le programme devra vérifier que le fichier à l'entrée utilise bien 16 bits pour représenter une valeur avant de faire la modification.

Par ailleurs, la compagnie aimerait aussi avoir une idée de la qualité de fichiers modifiés dont elle dispose (voir le dossier « *Programme2* » du fichier « *medias-TP2.zip* »). Vous devrez évaluer le rapport signal à bruit (RSB) des fichiers modifiés par rapport au fichier original (« *App2_Original_Mono8bits.wav* »). Pour ce faire, vous devrez développer un second filtre audio pour calculer le SNR (Signal to Noise Ratio). Le fichier original servira de référence dans le calcul. Vous aurez donc deux petits programmes : un premier qui traite les fichiers originaux, et un second qui évalue le SNR des fichiers modifiés par rapport à l'original.

Exigences

L'entreprise conserve actuellement ses fichiers dans un format non compressé. Elle aimerait les conserver de cette façon après qu'ils aient été modifiés.

Votre filtre ne devrait pas gaspiller de la mémoire inutilement. Évitez de lire le fichier entièrement avant de le traiter. Optez plutôt pour la lecture du fichier en morceaux pour diminuer la mémoire consommée.

Les lecteurs communs par exemple, Windows Media Player, devront être en mesure de jouer le fichier. Tous les paramètres, par exemple, la durée du fichier,

devront être corrects. Si le lecteur audio affiche une durée d'une minute pour le fichier original (et que cette durée est correcte), le lecteur devrait afficher la même durée lorsque le lecteur fait jouer le fichier que votre programme a produit.

Lorsque vous évalueriez le SNR, vous donnerez le fichier original comme fichier de référence ainsi qu'une série de fichiers que vous avez modifiés. Vous conserverez le nom des fichiers et les valeurs de SNR dans une structure de données (double tableauSNR[]) afin de les trier à l'aide de la méthode de tri par insertion en ordre du meilleur SNR au pire SNR avant d'afficher les résultats. Le calcul du SNR s'effectue en appliquant l'équation suivante :

$$SNR = 10 * \log \left(\frac{\sum_{i=1}^{NbEch.} (Echantillon_i^{original})^2}{\sum_{i=1}^{NbEch.} (Echantillon_i^{original} - Echantillon_i^{modifié})^2} \right) dB$$

Architecture

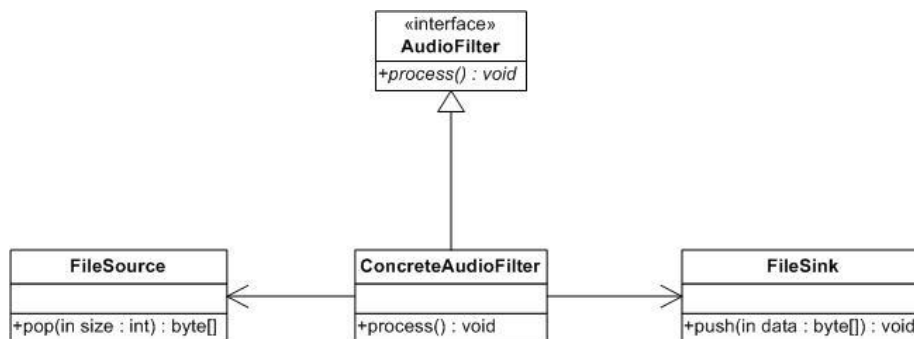


Figure 1 Diagramme de classes pour votre filtre particulier

Les tâches à compléter

- Créer la classe « ConcreteAudioFilter » (lui donner un nom approprié) et compléter la fonction « process ».
- Créer la classe « SNRFilter » et compléter la méthode « process ».

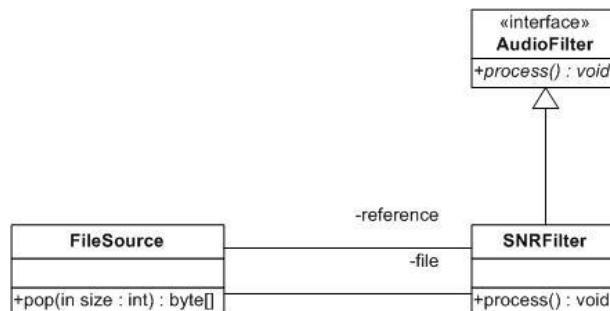


Figure 2 Diagramme de classe pour le filtre SNR

Exécution

Chacun de vos programmes devra accepter deux paramètres à l'appel : le nom du fichier d'entrée et le nom du fichier de sortie. Ainsi, si vous appelez vos applications depuis la console, vous devez respecter la syntaxique suivante :

```
java <programme1> <fichier d'entrée> <fichier de sortie>
```

```
java <programme2> <fichier de référence> <fichiers à analyser>
```

Dans Eclipse, vous pouvez spécifier les paramètres que vous désirez passer à votre application en suivant les étapes suivantes. Sélectionnez le menu *Project* → *Properties*. Puis, sélectionnez l'onglet « *Run/Debug Settings* ». Créez une nouvelle configuration de type « *Java Application* ». Appelez votre configuration comme désiré et cliquez sur l'onglet « (x) = *Arguments* ». Chaque ligne du cadre « *Program arguments* : » correspond à un paramètre de votre application. Ensuite, lorsque vous désirez exécuter votre configuration, cliquez sur le menu *Run* → *Run Configuration...* Sélectionnez la configuration que vous venez de créer.

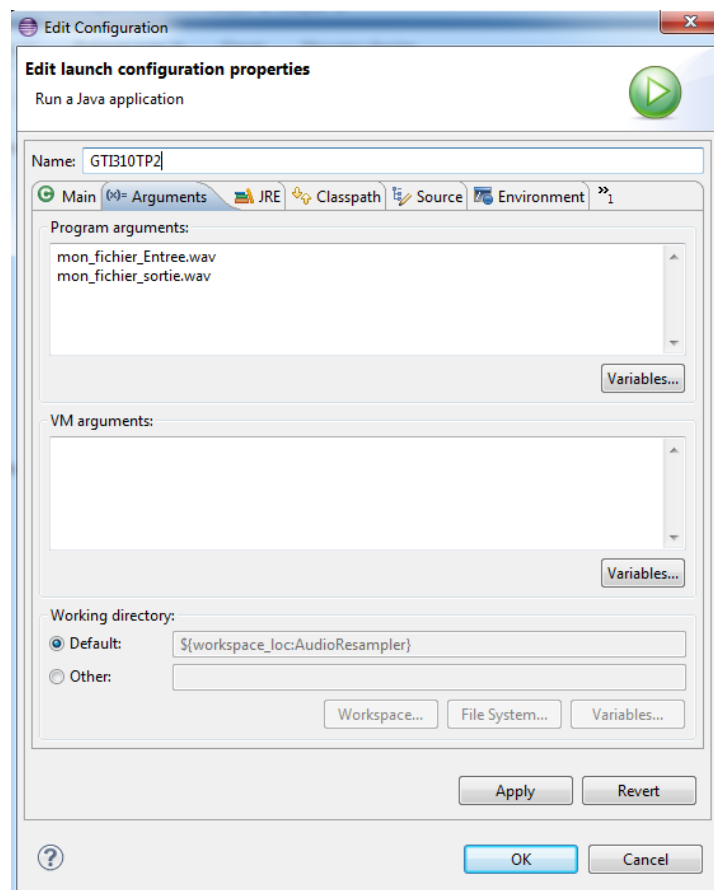


Figure 3 Configuration java pour passer des paramètres à votre application

Analyse de complexité

Vous devrez analyser (notation O) la complexité des méthodes de votre classe « AudioFilter ». Cette analyse devra se trouver dans les commentaires de l'entête des méthodes servant à la résolution de problème.

Résultats attendus pour l'application 2

Voici les SNR que vous devez obtenir :

Fichier audio	SNR en dB
App2Mod1AmpBaseMono8bits.wav	18.81
App2Mod2Delai30SamplesMono8bits.wav	18.87
App2Mod3Echo1SecMono8bits.wav	25.86
App2Mod4WhiteNoise1Mono8bits.wav (bruit de .1)	20.55
App2Mod5WhiteNoise2Mono8bits.wav (bruit de .2)	18.74
App2Mod6WhiteNoise3Mono8bits.wav (bruit de .05)	30.67
App2Mod7WhiteNoise4Mono8bits.wav (bruit de 0.01)	42.02
App2Mod8PinkNoiseMono8bits.wav	24.13

À remettre

- Un rapport de laboratoire
 - Voir le gabarit qui est disponible sur le site du cours.
- Le projet contenant votre code source.
 - **Ne pas remettre les fichiers audio utilisés ou produits.**

Critères d'évaluation

1. Rapport de laboratoire : 40 %
2. Programme : 60 %
 - a. Qualité du code : 15 %
 - b. Fonctionnement : 40 %
 - c. Analyse de complexité : 5 %

Date de remise

Au plus tard le 28 février 2016 à 13h30. Vous devez effectuer une remise électronique sur Moodle (code java et rapport de laboratoire en format .doc ou .docx).

Pénalité de retard

Les dates de remises des travaux doivent être respectées. Une pénalité de 10 % par jour ouvrable sera appliquée aux travaux qui ne sont pas remis à temps.

Plagiat et fraude

Les clauses du « Chapitre 10 : Plagiat et fraude » du « Règlement des études de 1er cycle » s'appliquent dans ce cours ainsi que dans tous les cours du département de génie logiciel et des TI.