

IFT232 MÉTHODES DE CONCEPTION ORIENTÉE OBJET

DEVOIR #2

Date limite de remise : le octobre 2014 à 23h59

Mode de remise : soumettez une archive se nommant tp2.zip via la commande turnin.

Il contiendra le code source du programme refactorisé, ainsi que le code des tests.

Veillez SVP inclure un projet eclipse complet (clic droit sur le projet, export, archive) pour faciliter la correction.

Veillez également inscrire les noms des coéquipiers dans l'en-tête du fichier MultimediaManager.java

Vous avez été engagé par NULLSOFT pour implémenter un système d'échange de documents multimédia P2P (peer-to-peer).

En particulier, vous aurez à implémenter un gestionnaire de contenu multimédia `MultimediaManager` multi-plateforme et facilement extensible.

- Les opérations possibles sont *start*, *pause*, *resume* et *stop*.
- Ce gestionnaire doit utiliser le « player » approprié (vidéo, MP3, WAV, etc.) pour la plateforme où il s'exécute (Apple, Windows, etc.).
- Ce gestionnaire contrôle aussi le nombre d'utilisation du fichier multimédia. Les contenus multimédias sont soit vendus ou gratuits (nombre illimité d'utilisation), soit loué pour un nombre déterminé d'utilisation. Une utilisation correspond soit à une écoute complète (i.e. la méthode « stop » du lecteur `MediaPlayer` a été invoquée), soit à une écoute incomplète non reprise, i.e. l'utilisateur a appuyé sur la fonction « pause » et n'a jamais repris l'écoute). Par exemple, les séquences suivantes correspondent chacune à une utilisation
 - START, PAUSE, RESUME, STOP
 - START, PAUSE, RESUME, PAUSE, RESUME, STOP
 - START, PAUSE, RESUME, PAUSE

Vous disposez comme point de départ des classes et interfaces décrites ci-dessous. Si des classes et des méthodes manquent, vous pouvez coder des « dummy », (mock objects, ou objets bidon).

Question 1 : Tests unitaires (50%)

Ecrire au moins 10 tests unitaires pertinents avec JUnit. Les tests doivent démontrer les fonctionnalités et les contraintes de `MultimediaManager` énoncées ci-haut. Attardez-vous à bien analyser quel est le fonctionnement externe (spécification) qui est recherché pour écrire vos tests. Vos tests devront évoluer avec la question 2 et encore démontrer les mêmes fonctionnalités.

Question 2 : Refactorisation (50%)

Refactoriser le programme en faisant disparaître les conditionnelles lorsque c'est possible; l'application du state pattern est fortement suggérée pour atteindre cet objectif.

Le code source est disponible avec cet énoncé sur le répertoire public.

Pour démarrer le devoir, créez un nouveau projet Eclipse. Ensuite, faites un clic droit sur le répertoire « src » de votre projet, puis choisissez « import... », ensuite, choisissez « general », « archive file » et trouvez le chemin du fichier jar. Le code source sera importé dans votre projet.

Pour utiliser JUnit, allez dans les propriétés du projet, dans la catégorie « java build path », sous l'onglet « libraries », utilisez le bouton « add library » et choisissez JUnit (version 4).