

IFT232 MÉTHODES DE CONCEPTION ORIENTÉE OBJET

DEVOIR #2

Date limite de remise : le 7 novembre 2012 à 23h59

Mode de remise : déposer un fichier .jar sur Moodle. Ce fichier se nommera *Devoir2-nom de famille.jar*. Il contiendra le code source du programme.

Le TP2 est la suite du TP1. Aussi pour le réaliser, vous devrez utiliser comme base de départ la solution du TP1.

Question 1 : Suivi de la lecture du document Multimedia (15 points)

L'unique instance de la classe `Afficheur` (singleton) permet de suivre sur la console l'exécution des documents multimédia. A l'aide des patrons de conceptions observateur et adaptateur, modifiez le programme pour faire en sorte que le `MultimediaManager` informe l'afficheur de son changement d'état. **Vous NE devez PAS modifier** la classe `Afficheur` pour implémenter cette fonctionnalité. Vous pouvez modifier le reste du programme.

Question 2 : Gestion des locations (15 points)

Dans la solution du TP1, la gestion des locations se fait dans la classe `MultimediaManager` à l'aide de la variable d'instance `utilisation` et des méthodes `canPlayOneMoreTime`, `incUtilisation` et `decUtilisation`. Peu importe qu'il s'agisse d'une location ou pas, le nombre de fois qu'un document est joué est comptabilisé. Cette information n'est pourtant pertinente que s'il s'agit d'une location. La classe `MultimediaManager` fait donc deux choses distinctes : s'occuper de faire jouer les documents multimédia et gérer le nombre de fois que l'on peut faire jouer un document loué.

Pour améliorer la modularité du code, vous devez séparer ces deux aspects. La classe `MultimediaManager` s'occupera de faire jouer les documents; la classe `LocationManager` s'occupera de gérer les éléments relatifs aux locations. A l'aide du patron observateur, vous devez donc extraire la gestion des locations de la classe `MultimediaManager` et faire en sorte que les opérations inhérentes à une location ne soit réalisées que s'il s'agit effectivement d'une location et que le nombre total d'utilisations permis n'a pas été dépassé.

Piste de solution: utiliser les interfaces `PropertyChangeListener` et `VetoableChangeListener`.

```

public class MultimediaManager {

    final static String VIDEO = ".mov";
    final static String MUSIQUE = ".mp3";

    final static private int CREATED = 1;
    final static private int PAUSED = 2;
    final static private int PLAYING = 3;
    final static private int STOPPED = 4;

    private int etat_;
    private int maximum_;
    private int utilisation_;
    private boolean location_;

    private String auteur_;
    private String titre_;
    private String categorie_; //MUSIQUE ou VIDEO
    private int taille_;
    private Object contenu_;

    private MediaPlayer player_;

    /**
     * s'il s'agit d'une location, la variable utilisation_
     * ne doit jamais être supérieure à maximum_
     *
     * s'il s'agit d'une location,
     * la variable d'instance maximum_ doit être supérieure à 0
     *
     * s'il s'agit d'une location,
     * la valeur de la variable d'instance utilisation_ correspond
     * à la différence entre le nombre d'invocations
     * des méthodes pause() et stop() et
     * le nombre d'invocations de la méthode resume()
     */

    public MultimediaManager( boolean location, int maximum,
                             String auteur, String titre,
                             String categorie, int taille, Object contenu) {

        location_ = location;
        maximum_ = maximum;

        auteur_ = auteur;
        titre_ = titre;
        categorie_ = categorie;
        taille_ = taille;
        contenu_ = contenu;

        utilisation_ = 0;
        etat_ = CREATED;
    }
}

```

```

public void start() {

    // cette méthode ne peut être invoquée
    // que pour démarrer la lecture du document multimédia

    if (utilisation_ == maximum_) {
        return;
    }

    if (etat_ == CREATED) {
        if (categorie_ == MUSIQUE) {
            player_ = new iTunes();
            player_.play(titre_, auteur_, categorie_, taille_, contenu_);
            etat_ = PLAYING;
        }
        if (categorie_ == VIDEO) {
            player_ = new QuickTime();
            player_.play(titre_, auteur_, categorie_, taille_, contenu_);
            etat_ = PLAYING;
        }
    }
    if (etat_ == STOPPED) {
        if (categorie_ == MUSIQUE) {
            player_.play(titre_, auteur_, categorie_, taille_, contenu_);
            etat_ = PLAYING;
        }
        if (categorie_ == VIDEO) {
            player_.play(titre_, auteur_, categorie_, taille_, contenu_);
            etat_ = PLAYING;
        }
    }
    // l'état de l'instance est PLAYING
}

public void pause() {

    //l'instance doit être dans l'état PLAYING
    //pour que cette opération soit valide

    if (categorie_ == MUSIQUE) {
        player_.pause(titre_, auteur_, categorie_, taille_, contenu_);
        if (location_) {
            utilisation_ = utilisation_ + 1;
        }
        etat_ = PAUSED;
    }

    if (categorie_ == VIDEO) {
        player_.pause(titre_, auteur_, categorie_, taille_, contenu_);
        if (location_) {
            utilisation_ = utilisation_ + 1;
        }
        etat_ = PAUSED;
    }
    // l'état de l'instance est PAUSED
    // le nombre d'utilisation est incrémentée de 1
}

```

```

public void resume() {

    // cette opération n'est valide qu'après une pause

    if (location_) {
        utilisation_ = utilisation_ - 1;
    }
    if (categorie_ == MUSIQUE) {
        player_.play(titre_, auteur_, categorie_, taille_, contenu_);
        etat_ = PLAYING;
    }
    if (categorie_ == VIDEO) {
        player_.play(titre_, auteur_, categorie_, taille_, contenu_);
        etat_ = PLAYING;
    }
    // l'état de l'instance est PLAYING
    // le nombre d'utilisation est décrémentée de 1
}

public void stop() {

    //le contenu doit jouer pour que cette opération soit valide

    if (categorie_ == MUSIQUE) {
        player_.close(titre_, auteur_, categorie_, taille_, contenu_);
        if (location_) {
            utilisation_ = utilisation_ + 1;
        }
        etat_ = STOPPED;
    }

    if (categorie_ == VIDEO) {
        player_.close(titre_, auteur_, categorie_, taille_, contenu_);
        if (location_) {
            utilisation_ = utilisation_ + 1;
        }
        etat_ = STOPPED;
    }

    // l'état de l'instance est STOPPED
    // le nombre d'utilisation est incrémentée de 1
}

}

public interface MediaPlayer {
    void pause(String titre, String author,
               String category, int length, Object contents);
    void close(String titre, String author,
               String category, int length, Object contents);
    void play(String titre, String author,
               String category, int length, Object contents);
}

public class iTunes implements MediaPlayer { /* code de la classe */ }
public class QuickTime implements MediaPlayer { /* code de la classe */ }

```