

Guide du développeur

LOG8430: TP1

Par:

Guillaume Rivest (1438615)

Clément Duffau (1699323)

Mohammed Benbachir ()

Contents

Présentation du guide du développeur	2
Présentation du logiciel	2
Explication de l'architecture	2
Description sommaire des classes	2
Diagramme de classes.....	2

Présentation du guide du développeur

Ce guide est à l'attention des développeurs. Il y est expliqué les choix de d'architecture et l'implémentation.

Présentation du logiciel

Le logiciel propose un outils de sélection de fichier ou dossier et 3 commandes applicables sur ce fichier ou dossier. Certaines de ces commandes sont applicables sur un fichier ou dossier, d'autres exclusivement sur un des deux types précédents.

L'interface proposée permet de lancer ces 3 commandes à l'aide des 3 boutons ainsi que voir leurs effets. Suite à la sélection d'un fichier, un panneau indique dans le cas d'un fichier, son nom tandis que pour un dossier, la liste des fichiers ou dossiers qu'il contient

Explication de l'architecture

L'architecture choisie pour la partie métier de l'application est basée sur un patron de conception commande. Le choix a été fait de privilégier ce schéma de conception car il permet d'avoir une commande générique qui agrège un invocateur chargé de récupérer les paramètres des commandes et le modèle de données manipuler. Ne reste plus qu'à étendre cette interface pour créer des commandes concrètes et leurs actions.

L'architecture choisie pour la partie interface homme-machine est basée sur le patron de conception MVC. Le choix a été fait de privilégier ce schéma de conception car il permet de bien séparer les différentes partie de l'IHM. Une partie pour le modèle de données (chargement dans un base de données, structures de données, ...). Une autre partie qui gère la vue en total indépendance du modèle. Un contrôleur qui manipule et créer les interaction avec le modèle de données et la vue.

Description sommaire des classes

« Invocateur » est responsable des entrées utilisateurs.

« Commande » est l'interface qui définit les méthodes génériques d'une commande. Ici, on se restreint à `executer()` mais on peut très bien imaginé rajouter `desexecuter()` et `estReversible()` pour étendre les fonctionnalités du logiciel.

« CommandeFichier » est une classe abstraite chargé de représenté une commande générique sur un fichier ou dossier. Elle sert de modèle de données dans notre cas

« CommandeVide1 », « CommandeVide2 » et « CommandeVide3 » représente les 3 commandes de notre système.

« CommandeFrame » est la vue de notre application

« CommandeManager » est le controleur de notre application

Diagramme de classes

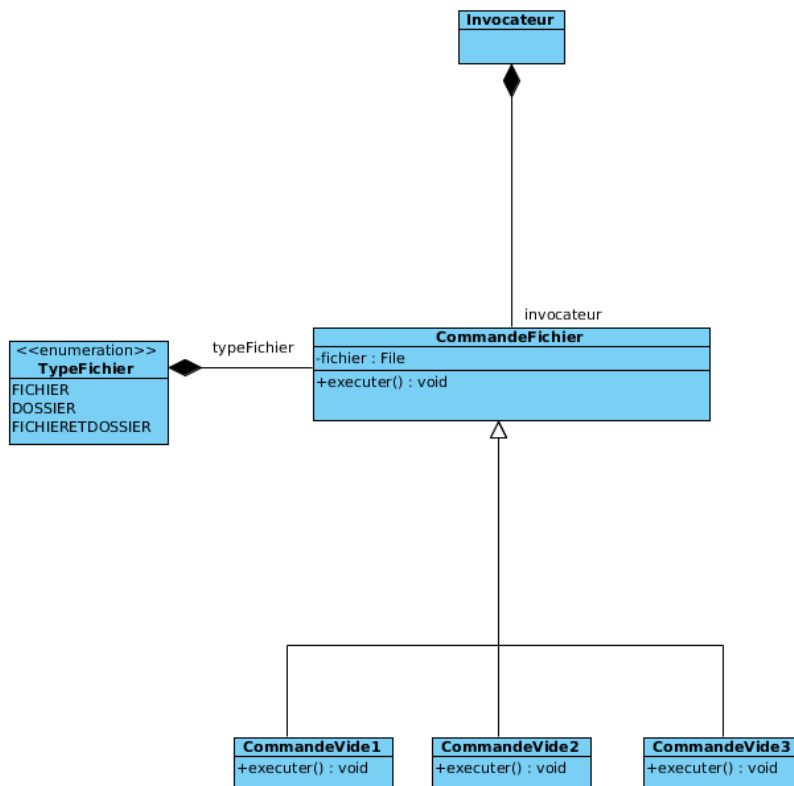


Figure 1 : Diagramme de classes de la Partie 1

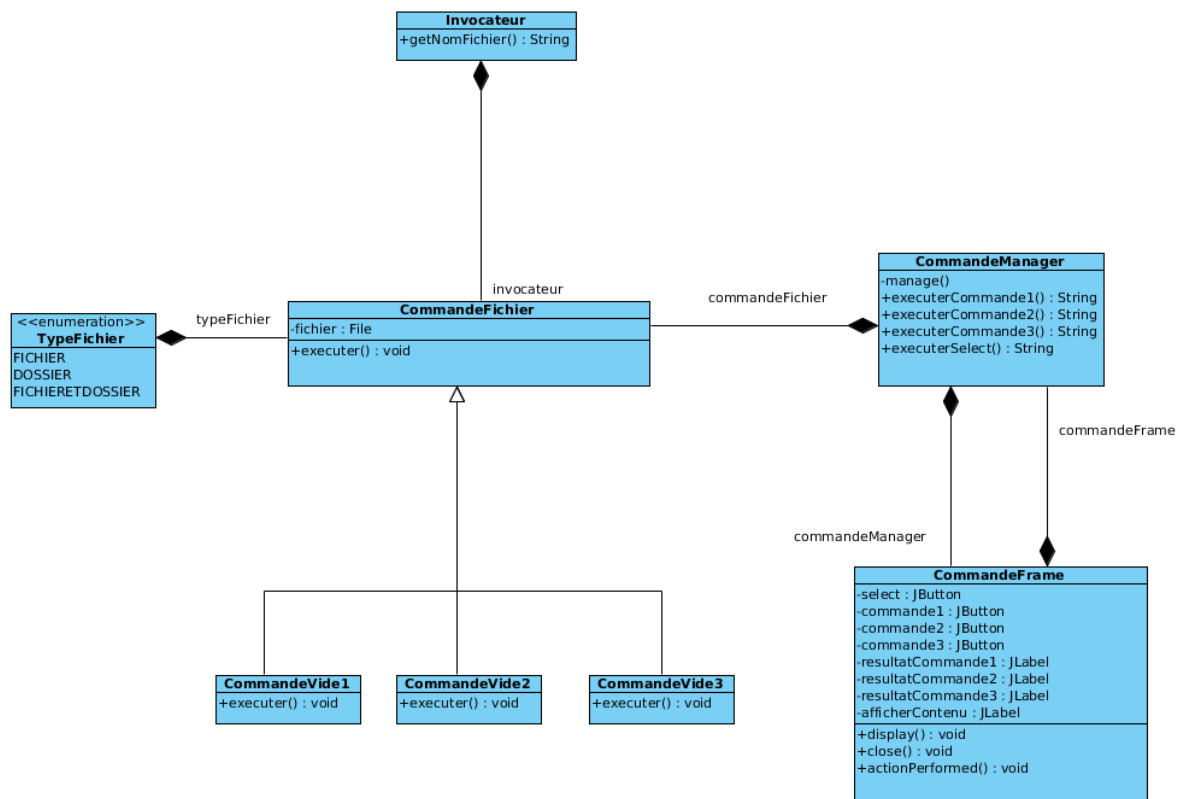


Figure 2 : Diagramme de classes de la Partie 2