**TD 7 - TP télécommande - Ajout d'autres appareils :**

**Chouadra Rayan S2D.**

Dans le cadre de ce TP, j’ai travaillé sur l'extension de la classe Télécommande pour lui permettre de manipuler non seulement des chaînes Hi-Fi mais également d'autres types d'appareils tels que des télévisions et des lampes. L'objectif principal était d’évité les copier-coller.

**1. Manipulation de Chaîne Hi-Fi**

Nous avons modifié la classe Télécommande pour qu'elle puisse également manipuler des chaînes Hi-Fi en plus des fonctionnalités déjà existantes. Pour ce faire, nous avons ajouté une classe Hifi représentant une chaîne Hi-Fi avec des méthodes pour allumer, éteindre et ajuster le volume.

**2. Réponses aux questions posées**

**2.1 Diagramme de classe**

Nous avons créé un diagramme de classe simplifié illustrant le fonctionnement de la classe Télécommande, intégrant la manipulation des chaînes Hi-Fi.

Télécommande :

---------------------------------

- appareils : List<Appareil>

---------------------------------

+ ajouterAppareil(appareil: Appareil): void

+ allumerAppareil(index: int): void

+ eteindreAppareil(index: int): void

+ changerCanal(index: int, canal: int): void|

+ augmenterVolume(index: int): void

+ diminuerVolume(index: int): void

+ afficherEtat(): void

Appareil :

---------------------------------

- nom: String

- etat: boolean

---------------------------------

+ allumer(): void

+ eteindre(): void

+ afficherEtat(): void

Hifi | | Television

---------------- -----------------

- son: int | | - canal: int

---------------- -----------------

+ allumer(): void

+ eteindre(): void

+ augmenterSon(): void

+ diminuerSon(): void

+ afficherEtat(): void

**2.2 Modification du code de v1.0**

Les modifications apportées à la version v1.0 pour gérer les chaînes Hi-Fi comprenaient principalement l'ajout de méthodes et de fonctionnalités spécifiques à la classe Hifi. J’ai réussi a comparé mes deux versions avec la commande diff de git. J’ai pour la class télécommande moins de ligne de 77 a 45 environ car j’ai simplifier la classe pour quelle ajoute à chaque fois un nouvelles appareils.

**2.3 Ajout d'une télévision**

En supposant que la classe Télévision fonctionne de manière similaire à la classe Hifi, j’ai fait un diagramme de classe permettant de manipuler des télévisions avec la même télécommande, sans écrire de code.

**2.4 Copier-coller**

J’ai évité les copier-coller dans la classe Télécommande afin de garantir la maintenabilité du code. Si du code avait été copié-collé et qu'il comportait un bug, cela aurait été reproduit dans plusieurs endroits, ce qui rendrait la correction plus difficile. De même, toute modification des indices ou des méthodes des appareils aurait nécessité des modifications répétées.

**2.5 Relation de dépendance**

Dans mon diagramme de classe, les flèches représentant les associations vont de la télécommande aux appareils (par exemple, Hifi, Télévision), ce qui signifie que la télécommande dépend des appareils pour fonctionner. Si la classe Hifi n'avait plus de méthodes allumer et éteindre mais plutôt augmenter Son ou baisser Son, cela nécessiterait des modifications dans la télécommande pour s'adapter à ces nouvelles méthodes.

Appareil :

---------------------------------

- nom: String

- etat: boolean

---------------------------------

+ allumer(): void

+ eteindre(): void

+ afficherEtat(): void

Télécommande :

---------------------------------

- appareils : List<Appareil>

---------------------------------

+ ajouterAppareil(appareil: Appareil): void

+ allumerAppareil(index: int): void

+ eteindreAppareil(index: int): void

+ changerCanal(index: int, canal: int): void|

+ augmenterVolume(index: int): void

+ diminuerVolume(index: int): void

+ afficherEtat(): void

Television

----------------

- canal: int

----------------

+ allumer(): void

+ eteindre(): void

+ changerCanal(canal: int): void

+ afficherEtat(): void

**3. Démarche**

**3.1 Écriture d'un descriptif de l'application**

La classe Télécommande permet de contrôler différents appareils électroniques tels que des chaînes Hi-Fi, des télévisions et des lampes. Elle offre des méthodes pour allumer, éteindre et ajuster les paramètres de ces appareils via une interface utilisateur simplifiée.

**3.2 Diagramme de classe**

J’ai fait un nouveau diagramme de classe permettant à la télécommande de manipuler des lampes et des chaînes Hi-Fi sans avoir besoin d'ajouter des méthodes dans Télécommande.

**3.3 Développement**

Mon code correspond à cette nouvelle conception, en ajoutant la prise en charge des lampes en plus des chaînes Hi-Fi.

**3.4 Retour sur les questions précédentes**

La nouvelle conception répond aux questions posées précédemment en évitant les copier-coller en anticipant les évolutions futures telles que l'ajout de nouveaux types d'appareils.

**3.5 Ajout d'un nouvel appareil**

Pour ajouter un nouvel appareil manipulable par la télécommande, tel qu'une télévision, il suffit de créer une nouvelle classe représentant ce type d'appareil avec des méthodes appropriées pour le contrôler (par exemple, allumer, éteindre, changer de chaîne) et de mettre à jour le diagramme de classe pour refléter cette modification.

**Conclusion**

Ce TP m’a permis de mettre en pratique des concepts de conception logicielle tels que la modélisation à l'aide de diagrammes de classe, la réutilisation de code et l'anticipation des évolutions futures.