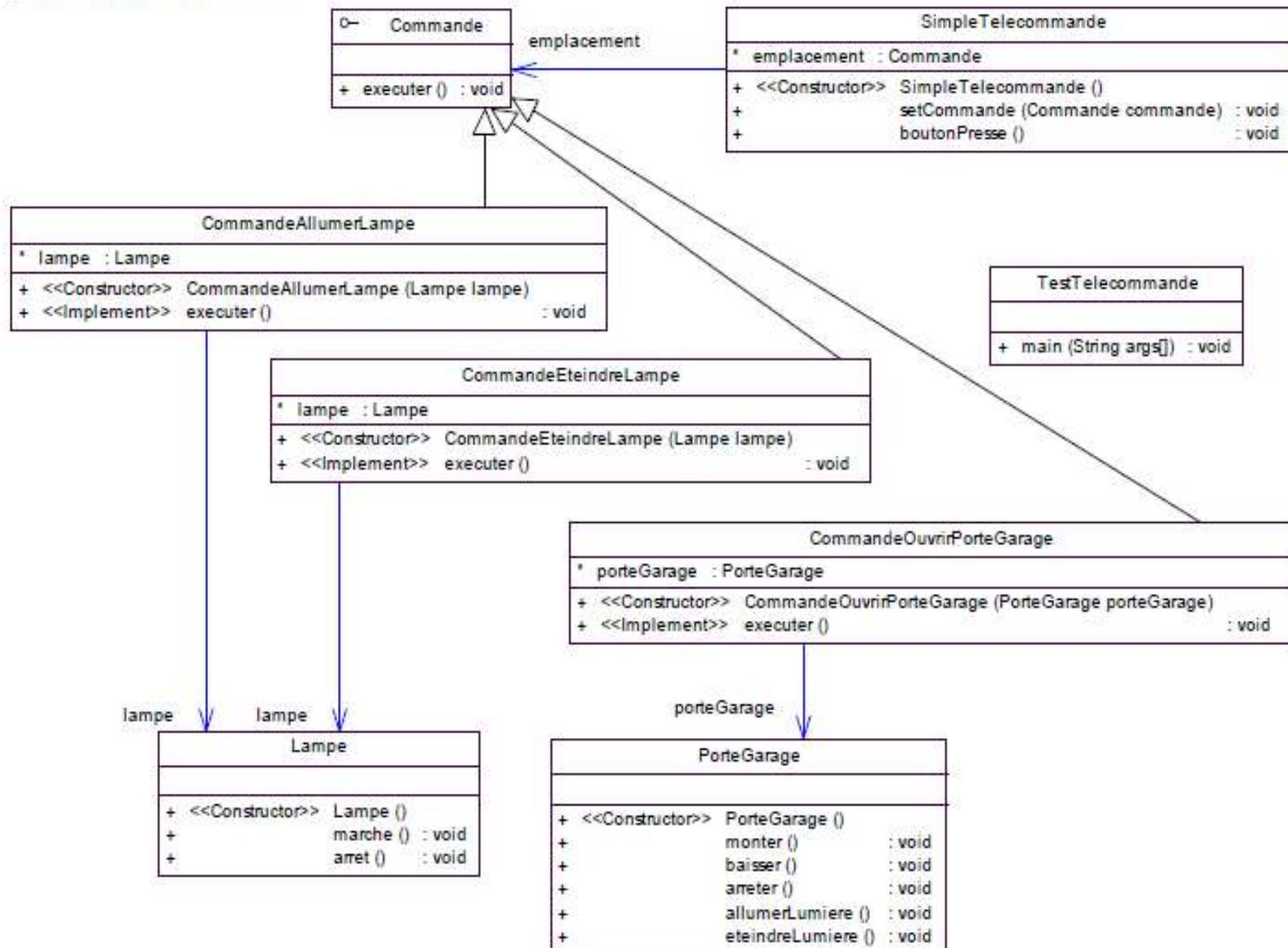
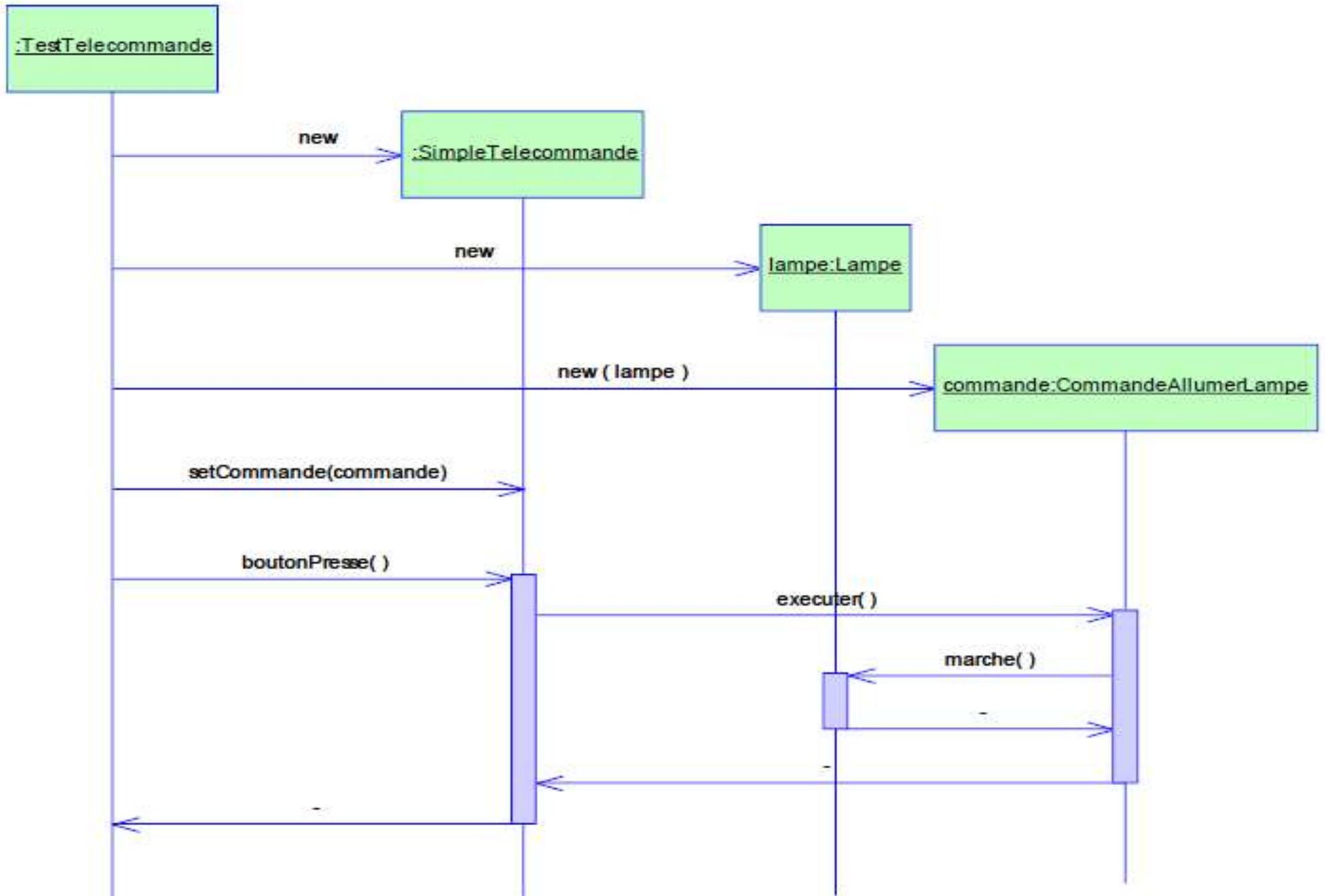


Diagramme de classe



-
- Le principe est que les objets à commander existent déjà avec leurs méthodes publiques (la lampe et la porte de garage). On veut les manipuler via la télécommande. On crée donc une interface de commande avec la méthode « exécuter » et des classes de commandes spécifiques. Si un nouvel objet arrive (un ventilateur) on ajoutera les classes de commandes spécifiques du ventilateur. La télécommande ne change pas. Et quand on fabrique une télécommande concrète, on lui donne les commandes qu'on veut.



-
- Le diagramme de séquence montre :
 - La création de la télécommande : jusqu'au setCommande. On crée la télécommande, puis les objets à manipuler, puis les commandes qui permettent de manipuler les objets. Ces commandes sont settées à la télécommande.
 - L'utilisation de la télécommande : boutonPresse() : cette utilisation est générique : selon la méthode associée au bouton, on aura tel ou tel comportement. Comme dans le DP Stratégie, on sette une méthode à la télécommande via un objet. Mais la télécommande accède à l'objet manipulé via une interface (l'interface Commande) qui définit la méthode exécuter, méthode qui est ensuite implémentée spécifiquement selon la commande et l'objet concerné.

-
- On doit programmer une télécommande qui prend en charge 10 boutons ON et OFF.
 - Les classes d'objets télécommandés sont fournies. Dans un premier temps, on n'utilise pas la classe ventilateur. La classe télécommande est fournie.
 - Vous devez coder le main avec tous les cas ON et OFF possible et le pattern commande. La classe `CommandeAllumerLampe` est donnée en exemple.
 - On précise que pour le Jackusi, OFF veut dire : `refroidir()`; puis `eteindre()`; et ON veut dire : `allumer()`; puis `chauffer()`; puis `bouillonner()`;
 - Pour la Stereo, ON veut dire : `marche()`; `setCD()`; `setVolume(11)`;
 - Pour le garage, ON veut dire `monter()` et OFF baisser.
 - 2) vous devez gérer le ventilateur qui peut être allumé en vitesse lente, moyenne ou rapide. Le principe de fonctionnement est qu'on utilise un bouton ON pour chaque vitesse et que le bouton OFF associé éteindra dans les 3 cas.
 - La classe ventilateur est fournie.
 - 3) Vous devez gérer l'annulation des commandes. L'annulation devra permettre de revenir à l'état antérieur pour toutes les commandes. La classe `CommandeAllumerLampe` est donnée en exemple.
 - 4) vous devez gérer une macro commande qui permet de tout allumer et de tout éteindre. Son principe est donné dans le diagramme de classe ci-dessous.

