

## **TP3 : Travailler sur le contrôle commande**

### **Architecture du système**

---

L'objet de cette séance est de vous faire travailler sur le contrôle commande. Pour ce faire, nous vous proposons la démarche ci-dessous :

Concevoir :

- Créer une interface `IControlCommand` afin de spécifier les méthodes que va offrir votre CC (enregistrer les requêtes et surveiller l'ascenseur). Valider cette interface avec votre enseignant.
- Créer une classe vide qui implémente l'interface en question et prévoir l'implémentation des méthodes.
- A ce stade, vous devez avoir un ordonnanceur fonctionnel et testé (voir TP précédent).
- Modifier le constructeur du CC pour injecter une instance de votre ordonnanceur dans le CC.
- Revoir les méthodes du CC qui assurent l'enregistrement des requêtes et construire un test unitaire pour les valider.
- Modifier à nouveau le constructeur afin d'injecter dans le CC une instance de `IElevator`.
- Prévoir une méthode `checkAndprocess` dont le but est de tester l'état de l'ascenseur afin de prendre éventuellement une décision (pour l'instant elle est vide).

### **La requête « aller à l'étage N »**

---

- Améliorer la méthode `checkAndprocess` pour prendre en compte les cas suivants :
  - L'ascenseur est arrêté portes fermées à l'étage  $E$  et il existe une requête à l'étage  $N$  (avec  $N > E$ )
  - Le cas symétrique avec  $N < E$
  - Le cas où  $E = N$
  - L'ascenseur est à la montée, il vient de franchir l'étage  $N$  et il existe une requête à l'étage  $N+1$ .
  - Le cas symétrique à la descente pour  $N-1$ .
- Compléter ces cas pour vous assurer que `checkAndprocess` a toujours un cas à appliquer (pour éventuellement ne rien faire). Générer une exception si cette règle n'est respectée.
- Vérifier par plusieurs tests unitaires le bon fonctionnement de votre CC. Conseil : vous devriez tester chaque cas en améliorant votre test unitaires cas par cas.

### **Les autres requêtes**

---

- En adoptant la même démarche (ajouter à `checkAndprocess` des cas exclusifs à prendre en compte), traiter les autres requêtes une par une avec, à chaque fois, la validation par un test unitaire.