

1 HomeAuto

Il s'agit de programmer un système à partir des *Tests unitaires* fourni.

Une fois votre travail terminé, faites dans l'ordre (c'est important!) les opérations suivantes:

1. Sauvegardez vos dernières modifications dans l'IDE.
2. Fermez l'IDE.
3. Ajoutez votre nom de famille comme préfixe du répertoire de votre solution.
Exemple: `Hurni-HomeAuto`
4. Compressez en ZIP ce répertoire
5. Transmettez cette archive au surveillant selon ses indications

1.1 Contexte

La classe `House` fait partie d'un système de domotique (fictif). Les fonctionnalités que vous devez implémenter gèrent un ensemble d'interrupteurs.

Chaque interrupteur a un *Nom* et un *Etat* (on/off).

Les fonctionnalités suivantes sont demandées:

- `AddSwitch(name)`: Ajoute un interrupteur à l'ensemble (il est éteint)
- `SwitchOn(name)`: Enclenche l'interrupteur correspondant à `name`
- `SwitchOff(name)`: Déclenche l'interrupteur correspondant à `name`
- `SwitchOnAll()`: Enclenche tous les interrupteurs
- `SwitchOffAll()`: Déclenche tous les interrupteurs
- `SwitchedOnCount`: Nombre d'interrupteurs enclenchés
- `SwitchedOffCount`: Nombre d'interrupteurs déclenchés

2 Etapes

2.1 Compilation réussie

Le code des tests utilise des membres de `House` qui ne sont pas encore défini.

Ajoutez tous ces membres afin que la compilation passe. Ceci permettra de lancer les tests et de les voir passer à rouge.

2.2 Tests en vert

Ajoutez le code nécessaire dans les membres de `House` afin de faire passer les tests à vert. Pour ceci basez vous sur le contexte explicité au point 1.

Vous avez le droit d'ajouter tout ce qui vous semble utile comme des membres supplémentaires ou d'autres classes.

La seule contrainte est que vous ne devez **pas toucher** le code des tests.