Classe Sensors

Le constructeur prend en paramètre un nombre qui représente le nombre de capteur. Ainsi l'attribut stateSensors est initialisé, toutes les valeurs sont initialisées à False.

Attribut :

- stateSensors : contient l'ensemble des capteurs et leurs état actuel.

Méthode :

- setState(dictionary<String, Boolean>) : permet de changer l'état d'un ou plusieurs capteurs.
- addName(String nameSensor) : permet de rajouter un capteur dans le dictionnaire, la valeur associée à ce capteur est False.
- getChange() : cette fonction vérifie toute les 7 secondes si un ou des capteurs changent d'état, si c'est le cas elle renvoie le dictionnaire contenant le nom de les capteurs ainsi que leurs états.
  getChange() va aussi modifier l'état des led concernées.

Classe abstraite *Observer* :

Attribut :

- subject : objet Sensor.

Méthode

- update() : méthode qui doit être implémentée de sorte à modifier l'état d'un observateur.

Classe Parking

Attribut :

- code : représente le code fournit par le client.
- stateSensors : représente le même dictionnaire que celui contenu dans la classe Sensors. Il est initialisé de la même manière. A chaque fois que la fonction getChange() de la classe Sensors retourne quelque chose, il est mis à jour.

Méthode :

- checkCode() : renvoie True si le code est bon et False dans le cas contraire.
- setCode(String code) : initialisation du code.
- numberFreeSpaces() : calcule le nombre de place disponible dans le parking.
- update() : met à jour le dictionnaire dans les attributs.
- openGate() : ouvre la barrière.