

Objectifs :

Ecrire un programme d'aide à la gestion d'un bloc d'ascenseurs

Nous vous conseillons d'écrire la classe **Test** en même temps que les autres. Par contre nous exigeons que vous écriviez les méthodes demandées dans l'ordre proposé. Et comme de coutume, vous préviendrez votre professeur dès que vous avez écrit une méthode. **Bon travail.**

Ascenseur

Ecrivez la classe **Ascenseur** dont les attributs sont privés et sont ;

étageCourant : entier, l'étage auquel se trouve l'ascenseur,

étageDemandé : entier, l'étage auquel l'ascenseur devra se rendre s'il est demandé, et s'il n'est pas demandé, *étageDemandé* vaudra *étageCourant*,

estDemandé : booléen, **true** si l'ascenseur est demandé,

et comme méthodes ;

- un seul constructeur qui initialise *étageCourant* et *étageDemandé* à la même valeur passée en paramètre et *estAppelé* à **false**,
- **boolean estArrivé()** ; retourne **true** si l'*étageCourant* est le même que l'*étageDemandé*,
- les 3 *assesseurs* qui renseignent sur les valeurs des attributs (*getteurs*),
- par contre on ne peut pas "poser" un ascenseur à l'étage que l'on veut : donc pas de "setteurs". Pour changer la position de l'ascenseur d'un étage, vous écrirez les 2 méthodes qui suivent ;
- **void monte()** ; qui incrémente l'*étageCourant* de l'ascenseur de **1** étage (un ascenseur ne peut pas monter 5 étages en une seule fois, mais en 5 fois 1 étage) et vérifie ensuite s'il n'est pas arrivé. Si l'ascenseur est arrivé, l'attribut *estAppelé* passe à **false**.
- **void descend()** ; idem que monte mais en décrémentant *étageCourant*,
- on ne peut appeler un ascenseur qu'en spécifiant l'étage demandé : donc on aura pour dernier setteur **void estAppelé(int étageDemandé)** ; change les attributs *estAppelé* à **true** et *étageDemandé* à la valeur du paramètre,
- **String toString()**

Exemples de retour : // L'ascenseur se trouve à l'étage 1. Il est libre.
// L'ascenseur se trouve à l'étage 1. Il va à l'étage 5.

GestionnaireAscenseur

Ecrivez la classe **GestionnaireAscenseur**. Cette classe va gérer plusieurs ascenseurs ; à chaque appel, elle enverra celui qui est le plus prêt, etc... Les attributs de cette classe sont privés et sont ;

NBASCENSEURS : le nombre d'ascenseurs gérés par le gestionnaire

ETAGEMAX : le plus haut étage que peuvent atteindre les ascenseurs,

ETAGEMIN : ???,

ascenseurs : un tableau d'*Ascenseur*(s),

et comme méthodes ;

- un seul constructeur, qui initialise tous les attributs, notamment *ascenseurs*,
- **boolean estEtageValide()** ; retourne **true** lorsque qu'un étage est demandé, et qu'il est plus petit que *ETAGEMAX* et plus grand que *ETAGEMIN*,
- **int indiceAscenseurLibre(int étageDemandé)** ; retourne l'indice de l'ascenseur **libre** le plus proche de l'étage demandé (paramètre) si l'étage demandé est valide, **-1** sinon,
- **boolean appelAscenseur(int étageDemandé)** ; retourne **false** si aucun ascenseur **libre** n'a pu être appelé, sinon fait monter ou descendre l'ascenseur libre le plus proche de l'étage demandé jusqu'à ce qu'il arrive à l'étage demandé, et retourne **true**
- **String toString()**

Exemple de retour : // Le bloc comporte : 2 ascenseurs
// _____
// L'étage le plus bas est -1
// L'étage le plus haut est 5
// _____
// L'ascenseur se trouve à l'étage 1. Il est libre
// L'ascenseur se trouve à l'étage 4. Il est libre

Vous testerez dans cette classe chacune des méthodes demandées.
Nous souhaitons des tests de la forme de ceux demandés au projet 1, toutefois nous nous rendons bien compte qu'une telle classe de tests prend du temps, du coup écrivez-la comme dans le projet 1 si vous en avez le temps, sinon affichez directement vos réponses.
Un bonus sera accordé aux étudiants qui auront écrit la classe Test comme dans le projet1.... Si, bien sûr, elle est bien écrite.