

Modélisation objets et Java - TP n°3

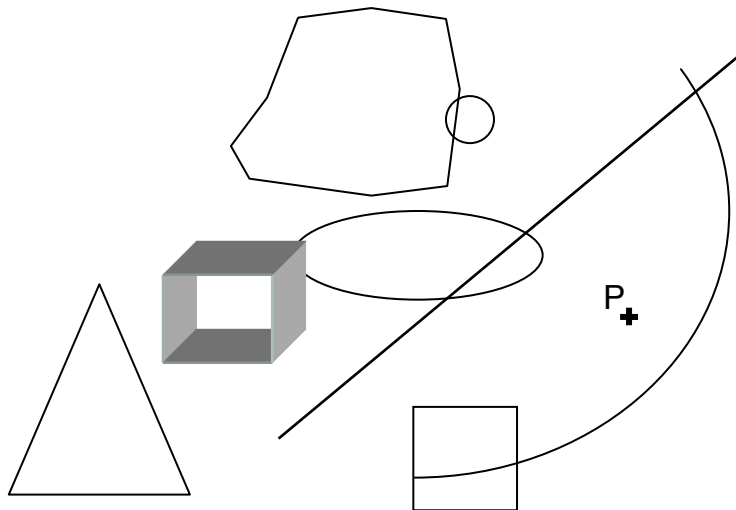
Exercice 1

Les machines électriques peuvent se classer en machines à courant alternatif (ca) et en machines à courant continu (cc). Certaines tournent en ca, d'autres en cc, et certaines dans les deux modes. Une machine en ca peut être synchrone ou inductive. Entre autres, on trouve de grands moteurs synchrones ou de petits moteurs inductifs, des moteurs universels ou des moteurs à aimants permanents. La plupart des moteurs des appareils ménagers sont des machines à induction ou des moteurs universels. Les moteurs universels sont en principe utilisés là où une vitesse élevée est nécessaire comme dans les mixeurs ou les aspirateurs à vide. Ils tournent sur des courants alternatifs ou continus. Des moteurs à aimants permanents sont employés fréquemment dans les jouets et ne tournent qu'en courant continu.

Proposez une hiérarchie des classes représentant les diverses catégories de machines.

Exercice 2

Proposez une hiérarchie des classes des éléments géométriques apparaissant dans cette image :



Exercice 3

Ajouter à la classe *Vector* une méthode attendant 2 *Vector* comme arguments et qui échange ces vecteurs.

Exercice 4

Ecrire une classe utilitaire *DoubleArray* qui gèrera des tableaux de tableaux de doubles. Cette classe définit les méthodes statiques suivantes :

- `display(double t[][]) :`
affiche les valeurs de t, ligne par ligne,
- `boolean regular(double[][] t) :`
test si le tableau est régulier, c'est-à-dire si les lignes sont de même longueur,
- `double[] lineSum(double[][] t) :`
retourne un tableau de doubles, chacun correspondant à la somme de la ligne,
- `double[][] sum(double[][] t1, double[][] t2) :`
si t1 et t2 sont réguliers et de même longueur, retourne leur somme ; sinon, retourne null.

Exercice 5

Ecrire une classe *Index* qui gère des numéros de téléphone. Ecrire d'abord une classe *Subscriber* qui contient le nom et le numéro d'un abonné.

Ajouter à la classe *Index* les méthodes suivantes :

- un constructeur à un argument de type *int* : le nombre maximum d'abonnés,
- `addSubscriber` pour ajouter un nouvel abonné, qui retourne faux si l'insertion échoue, vrai sinon.
- `getNumber(String name), getNbSubscribers(), getSubscriber(int i)`
- `getSortedSubscribers()` qui retourne un tableau d'abonnés, trié par ordre alphabétique.