

Faites un fichier différent par classe.

Pensez à tester vos méthodes et constructeurs au fur et à mesure dans une classe séparée des autres classes.

### Exercice 1

Un metteur en scène souhaite utiliser des robots dans une pièce de théâtre. Un groupe de robots aura un leader et tous les autres robots seront derrière lui en file indienne. Un groupe sera représenté par une liste chaînée.

On considère la classe Robot suivante :

```
public class Robot{
    private char nom;//lettre entre 'a' et 'z'
    private int energie;
    private String texte; //ce qu'il doit dire

    public Robot(char nom, String parole){
        this.nom = nom;
        // on donne une énergie entre 10 et 20
        this.energie = 10 + (int)(Math.random() * 11);
        this.texte = texte;
    }
}
```

1. Ajoutez une méthode `description` qui retournera une chaîne de caractères décrivant le robot du type "Robot <nom> dit <texte> quand il parle et a <energie> points d'énergie"
2. Ajoutez une méthode boolean `nomCorrect` qui vérifie qu'un robot a bien comme nom une lettre de l'alphabet minuscule.
3. Ajoutez des accesseurs permettant de récupérer les valeurs des différents attributs de Robot.

### Exercice 2

On considère les classes `Groupe` et `Cellule` suivantes :

```
public class Cellule{
    private Robot rob;
    private Cellule suivant;
}

public class Groupe{
    private Cellule leader;
}
```

Les méthodes demandées ici sont à faire en itératif, s'il y a lieu.

1. Ajoutez les constructeurs suivants :

- (a) dans la classe **Groupe**, un constructeur qui crée un groupe vide
- (b) dans la classe **Cellule**, un constructeur qui prend un argument de type **Robot** et un argument de type **Cellule** et un constructeur qui prend seulement un argument de type **Robot**.
- (c) Écrivez une méthode **void prendreDirection(Robot r)** qui teste si le robot **r** a un nom correct et si oui le met comme leader du groupe. L'ancien leader, s'il existait, est maintenant juste après le nouveau leader.
- (d) Écrivez une méthode **affiche()** qui affiche la description de tous les robots du groupe.
- (e) Écrivez une méthode **Groupe couperAPartirDe(char nom)** qui élimine tous les robots qui sont derrière le premier robot de nom **nom**. On retournera la liste de ceux qui ont été enlevés. **Indications** : On écrira dans **Cellule** une méthode **Cellule couperAPartirDe(char nom)** qui coupera la liste après la cellule contenant le robot cherché et qui retournera la cellule suivante ou null. Enfin, on ajoutera un constructeur **privé** dans **Groupe** qui prendra comme argument une cellule.
- (f) Écrivez une méthode **ajouteDebutant** qui ajoute un robot en fin de groupe, l'ajout n'a lieu que si le nom du robot est correct.
- (g) Écrivez une méthode **numerologie** qui donnera la somme des valeurs lettres des noms des robots mod 9. On suppose que tous les robots ont un nom correct ; la lettre 'a' vaut 1, la lettre 'b' 2, etc.

### Exercice 3 : facultatif

1. Écrivez une méthode **nom** qui donnera le nom du groupe constitué de la concaténation de tous les noms des robots pris dans l'ordre.
2. Les robots, doivent dire leur texte autant de fois que la position de leur nom dans l'alphabet. Par exemple, un robot de nom 'a' dira son texte une fois, un robot de nom 'b' dira son texte deux fois, etc. Faites une méthode **parle** dans **Robot** qui affiche le texte qu'il doit dire. Par exemple, si un robot qui s'appelle 'c' a comme texte "pomme", il dira "pommepommepomme". Après avoir parlé il perdra 10 points d'énergie, s'il lui reste moins de 10 points, il passe à 0.
3. Écrivez maintenant une méthode **faitParler** qui fait parler tous les robots du groupe.

### Exercice 4 : facultatif

Le metteur en scène veut maintenant plusieurs groupes de robots.

1. Définissez une classe qui représente un ensemble de groupes par un tableau.
2. Écrivez un constructeur qui crée un ensemble de groupe ; le nombre de groupes est donné en argument.
3. Écrivez une méthode **effectifs** qui retournera dans un tableau les effectifs de chaque groupe. Il faudra pour cela écrire des méthodes dans d'autres classes.