

## Travaux dirigés 2 Programmation orientée objet (Java) SMA5

### Exercice 1

- Définissez *une classe Point* permettant de manipuler un point du plan. Cette classe contiendra :
  - un constructeur,
  - les méthodes d'accès et de modification des coordonnées,
  - une méthode *deplacer* qui effectue une translation définie par les arguments,
  - une méthode *afficher* qui affiche les coordonnées d'un point,
  - une méthode *toString* qui transforme un point en une chaîne de caractères "(x,y)".
  - une méthode *distance* qui calcule la distance d'un point de l'origine
  - deux méthodes *rho* et *theta* qui renvoient les coordonnées polaires du point
  - une méthode *homothétie* qui applique au point une homothétie de centre (0; 0) et de rapport k,
  - une méthode *rotation*, qui applique au point une rotation de centre (0; 0) et d'angle a.

On donnera un exemple d'utilisation avec une méthode main

### Exercice 2

1. Ecrivez *la classe De* qui doit regrouper les attributs et les méthodes élémentaires permettant de gérer un dé :

- créer un dé (à sa création le dé une valeur),
- lancer d'un dé stockage de sa valeur. Nb: la méthode *random()* de la classe Math retourne un nombre au hasard entre 0 et 1.
- fournir la valeur du dé

2. Pour tester cette classe, vous écrirez un programme qui devra :

- créer un dé,
- effectuer un lancer,
- afficher le résultat du lancer,

3. Maintenant que la classe *De* est créée, vous allez pouvoir donner un peu plus d'interactivité au programme de test. Ecrivez un programme qui permet de lancer un dé plusieurs fois. On affichera le résultat après chaque lancer. Le nombre de lancer est passé en paramètre sur la ligne de commande.

4. Ecrivez un programme qui lance deux dés plusieurs fois. Le nombre de lancers est passé en paramètre sur la ligne de commande. Le résultat des dés est affiché à chaque lancer, et on dit si le joueur a gagné ou perdu (il gagne si la valeur des deux dés est la même).

### Exercice 3

- Définissez une classe *Individu* composée d'un *nom*, d'une *adresse*, et d'un *numéro de telephone*. Vous écrirez aussi les méthodes constructeurs, affectation et consultation des champs et affichage.
- Modifiez la classe *Individu* pour pouvoir :
  - afficher le nombre d'objets créés,
  - gérer la liste des individus créés,
  - afficher la liste des individus créés,

### Exercice 4

Ecrire une classe implantant une pile d'éléments.

- Comment représenter la pile vide ?
- Définir la **classe Pile**. Le constructeur de cette classe construira la pile vide.
- Définir une méthode permettant de tester si une pile est vide.
- Définir la méthode **empile** (ajoute un élément au sommet de la pile)
- Définir la méthode **dépile** (retourne le sommet et le retire de la pile)
- Définir la méthode **sommet** (retourne le sommet de la pile sans le retirer)
- On considère désormais qu'un élément de pile encapsule une valeur entière.
  - Comment représenter un élément de pile ?
  - Définir **la classe ElementPile** qui représente un élément d'une pile. Les attributs de cette classe seront privés. Définir les constructeurs et les accesseurs de cette classe.
  - Définir dans **la classe Pile** la méthode affiche qui affiche le contenu d'une pile.
- Tester la création d'une pile et sa manipulation en empilant puis dépilant divers éléments...