TP JAVA

Exercice 1

1) Il s'agit de définir une classe nommée Forme comportant deux méthodes abstraites :

double perimetre () et double surface() et une méthode non abstraite

void ContenantCarre(double surf) qui affiche le message suivant « cette forme peut contenir un carré de surface surf » si la forme a une surface au moins égale à celle d'un carré dont la surface est donnée par le paramètre surf et qui affiche sinon « cette forme ne peut pas contenir un carré de surface surf ».

2) Ecrire ensuite les deux classes Rectangle et Cercle héritant de cette classe Forme et définissant les deux méthodes abstraites de celle-ci.

La classe Rectangle a deux attributs entiers privés nommés longueur et largeur.

La classe Cercle possède un attribut privé de type double nommé rayon.

3) Tester la méthode Contenant Carre avec surf = 20 (dans la méthode main) en l'appliquant pour un rectangle de longueur = 2 et de largeur =1 puis avec surf =1 pour un cercle de rayon =1
4) Modifier ce programme pour qu'il affiche le message précédent (voir 1) en précisant en plus de quelle forme il s'agit (rectangle ou cercle) et en donnant ses dimensions (il ne s'agit pas de déplacer la méthode contenantCarre de sa classe).

Exercice 2

On vous demande d'écrire :

1) Une interface nommée Homme possédant une seule méthode :
   void identite () dont le rôle est d'afficher les informations concernant un homme.
2) Une classe Personne implémentant cette interface et possédant deux attributs privés nom et prénom de type String et un constructeur paramétré avec le nom et le prénom d'une personne.
3) Une classe Client héritant de la classe Personne et implémentant l'interface Homme avec un attribut privé supplémentaire nommé numéro de type entier et un constructeur paramétré en conséquence.
4) Une classe Peuple contenant deux attributs privés : un tableur d'Hommes nommé pays de capacité 100 et un entier nbHommes représentant le nombre d'hommes dans pays.

Cette classe contient également deux méthodes :

void naissance (homme h) : permet d'ajouter un homme dans pays

void explorer() : affiche l'identité de chaque homme dans pays

5) Tester les méthodes naissance () et explorer () après avoir placé dans pays des personnes et des clients.

Exercice 3

On veut construire un programme capable d'afficher et de déplacer des figures géométriques colorées dans un repère en deux dimensions.

Un graphique est un ensemble de figures affichables et déplaçables par translation. On veut pouvoir afficher des segments et des triangles. Ces deux éléments graphiques sont définis à l'aide de points.

Après une première analyse, on décide de modéliser l'application de la manière suivante :

– une classe Point représentera les coordonnées qui serviront à créer des figures

– une classe Segment sera défini par deux points et une couleur, codée par un entier positif.

– une classe Triangle sera défini par trois points et une couleur, codée par un entier positif.

– la classe Graphique est la classe principale contenant la méthode main et permettra d'afficher un ensemble d'objets graphiques.

Comme nous pensons devoir ajouter ultérieurement de nouveaux types de figures géométriques, nous décidons d'utiliser des interfaces pour décrire leurs comportements communs.

1. Après avoir étudié les différences et les points communs entre les trois premières classes, écrire une interface nommée drawable contenant les méthodes abstraites suivantes :
   - Méthode nommée setColor qui permet de choisir (modifier) une couleur.
   - Méthode nommée getColor qui retourne une couleur
   - Méthode nommée draw qui permet de dessiner (afficher les informations)
2. Ecrire une interface nommée Moveable contenant la méthode abstraite suivante :
   - Méthode nommée translate recevant en argument deux entiers et qui permet d'effectuer une translation.

3. Ajouter une classe Point contenant deux attributs privés (x et y) de type entier et les méthodes suivantes :
   - Constructeur qui reçoit en argument deux entiers et initialise x et y

- Constructeur qui reçoit en argument un point et initialise x et y
- Les mutateurs et les accesseurs (getters et setters)
- Méthode nommée toString qui retourne une chaîne de caractères décrivant le point.

Il est à noter que la classe Point implémentera l'interface Moveable. Les classes Segment et Triangle implémenteront les interfaces Moveable et Drawable.

4. Ajouter la classe Segment, caractérisée par deux points et une couleur privés, et contenant un constructeur qui reçoit en argument deux points et une couleur et initialise les attributs de la classe.
5. Ajouter la classe Triangle, caractérisée par trois segments et une couleur privés, et contenant un constructeur qui reçoit en argument trois segments et une couleur et initialise les attributs de la classe.
6. Ajouter la classe publique graphique contenant la méthode main() et permettant de créer et d'afficher un tableau de figures géométriques contenant un triangle et un segment.