

TP JAVA

DUREE : 1H30

Exercice 1 : Ecrire une classe **Rectangle** qui permet de calculer les dimensions d'un rectangle. On part du principe que chaque rectangle dispose d'une largeur et d'une longueur. Ces dimensions sont des entiers positifs dont l'unité est le mètre.

La classe dispose des méthodes suivantes :

- **Rectangle()** : un constructeur qui permet de créer un rectangle de dimensions 1 x 1
- **Rectangle (int longueur, int largeur)** : un constructeur qui permet de créer un rectangle de dimensions **longueur x largeur**
- **perimetre()** : qui retourne le périmètre du rectangle sur lequel il est appelé
- **surface()** : qui retourne la surface du rectangle sur lequel il est appelé
- **perimetre (int longueur, int largeur)** : une méthode de classe retournant le périmètre du rectangle dont les dimensions sont fournies en paramètre
- **surface(int longueur, int largeur)** : une méthode de classe retournant la surface du rectangle dont les dimensions sont fournies en paramètre
- Des **getters** et **setters** pour chacun de ses attributs

Ecrivez ensuite une classe **TestRectangle** permettant de tester la classe **Rectangle**.

Exercice 2 : Ecrire une classe **Vitesse** permettant de calculer la vitesse d'un coureur. Elle dispose de deux attributs d'instance que sont la distance (en mètre) et le temps (en seconde). Elle dispose également des méthodes publiques suivantes :

- **Vitesse (int distance, int temps)** : son seul et unique constructeur
- **metreParSeconde()** qui retourne la vitesse calculée en mètre par seconde
- **kilometreParHeure()** qui retourne la vitesse calculée en kilomètre par heure
- Des accesseurs et mutateurs pour chacun de ses attributs

Cette classe disposera également d'une méthode « main » qui permettra de tester les méthodes susmentionnées. Pour ce faire, le programme demandera à l'utilisateur la distance (en mètre), le temps (en seconde) et l'unité avant d'afficher la vitesse. On conviendra que l'unité ne pourra prendre que les valeurs MPS ou KPH pour respectivement dire « mètre par seconde » et « kilomètre par heure ». Si l'unité est en MPS, la vitesse calculée sera en mètre par seconde. Dans le cas contraire, elle sera en kilomètre par heure.

Exercice 3 : Ecrire une classe **Dessin** qui permet de dessiner certaines figures géométriques. Elle dispose des méthodes de classe suivantes :

- **rectanglePlein(int longueur, int largeur)** : permet de dessiner un rectangle plein avec les dimensions fournies
- **rectangleCreux(int longueur, int largeur)** permet de dessiner un rectangle creux avec les dimensions fournies
- **triangleIsocele(int nombreLignes)** qui est une méthode de classe permettant d'afficher un triangle isocèle formé d'étoiles et dont la hauteur correspond au nombre de lignes fourni.

Ecrire un programme qui permet de dessiner des figures suivant le choix de l'utilisateur. Les dimensions de ces dernières devront également être renseignées par l'utilisateur.



Rectangle (15 x 4) plein



Rectangle (15 x 4) creux



Triangle isocèle constitué de sept lignes

Exercice 4 : Créer une classe **Produit** qui représente les stocks de produits disponibles dans un supermarché. On part du principe que chaque produit dispose d'un nom, d'un prix unitaire et d'une quantité. Créer également une classe **GestionStock** qui dispose d'une liste de produits ainsi que du nombre de produits qu'il y'a effectivement en stock. Elle dispose également des méthodes suivantes :

- **GestionStock(int maxStock)** : un constructeur qui permet d'initialiser un tableau vide pouvant accueillir **maxStock** de **Produits**
- **void ajouterProduit(Produit p)** qui permet d'ajouter un produit dans le stock. L'ajout ne devra se faire que si la liste n'est pas pleine
- **void supprimerProduit(Produit p)** qui permet de supprimer un produit du stock (mettre **null** à la position occupée par le produit)
- **void listerProduit()** qui affiche la liste des produits, triée par ordre croissant du prix.

Ecrire un programme de test qui permet de saisir une liste de produits. A la fin de la saisie (le nombre de produits étant renseigné au préalable), la liste des produits saisis devra être affichée.